ASP.Net/MySQL Web Development Tutorials

Hadiya Harrigan’s Gold Award Project
Introduction
The purpose of this handbook is to introduce students to computer programming involving the MySQL database and Visual Studio/Visual Web Developer/Asp.net. This project was created as Hadiya Harrigan’s Gold Award Project. The handbook is divided into lessons. Each lesson presents a problem, walks students through solving the problem, and finally presents some individual or homework problems. Lessons may also include add-ons such as YouTube videos and links to other websites. Also, each lesson includes a pretest and a posttest to track student progress.

Why Use this Book?
This handbook uses the MySQL database and Visual Studio/Visual Web Developer/Asp.net. This combination is very hard to find useful information for. Therefore, this handbook solves this problem. Also, useful information is scattered all over the internet/ other resources. This handbook brings all of this information together. Lastly, all of the downloads for this course are free!
# Table of Contents

Please click the links to navigate.

Introduction ................................................................................................................. 2

Why Use this Book? ..................................................................................................... 2

Table of Contents ........................................................................................................ 3

Lesson Overview ......................................................................................................... 11

Topic List .................................................................................................................... 13

Downloads For This Course ..................................................................................... 16

  Purpose: .................................................................................................................... 16
  Downloading Visual Web Developer: ................................................................. 16
  Downloading MySQL Workbench: ................................................................. 16
  Downloading MySQL Community Server: ........................................................... 16
  Downloading MySQL ODBC Driver: ................................................................. 16

Unit 1 Section 1: Layout of a Web Page ................................................................. 18

  Purpose: .................................................................................................................... 18
  Overview: .................................................................................................................. 18
  Visual Web Developer Instructions: ................................................................. 20
  Problem Introduction: ......................................................................................... 25
  Problem Programming Steps: ........................................................................... 25
  Individual Problems: ............................................................................................ 32
  Tips: ......................................................................................................................... 32
  For More Information: .......................................................................................... 32

Unit 1 Section 1 Pre-Test ...................................................................................... 33

Unit 1 Section 1 Post-Test ..................................................................................... 34

Unit 1 Section 2: Buttons ....................................................................................... 35

  Purpose: .................................................................................................................... 35
  Overview: .................................................................................................................. 35
  Problem Introduction: ......................................................................................... 35
  Problem Programming Steps: ........................................................................... 36
  Individual Problem: ............................................................................................ 39
  Tips: ......................................................................................................................... 39
  For more information: .......................................................................................... 40

Unit 1 Section 2 Pre-Test ..................................................................................... 41
Unit 1 Section 2 Post-Test ................................................................. 42
Unit 1 Section 3: Simple Logic ........................................................... 43
  Purpose: .......................................................................................... 43
  Overview: ....................................................................................... 43
  Problem Introduction: ................................................................. 43
  Problem Programming Steps: ..................................................... 43
  Tips: ............................................................................................... 45
  For More Information: ................................................................. 46
Unit 1 Section 3 Pre-Test ................................................................. 47
Unit 1 Section 3 Post-Test ............................................................... 48
Unit 1 Section 4: Textboxes and Tables ......................................... 49
  Purpose: .......................................................................................... 49
  Overview: ....................................................................................... 49
  Problem Introduction: ................................................................. 49
  Problem Programming Steps: ..................................................... 49
  Tips: ............................................................................................... 53
Unit 1 Section 4 Pre-Test ................................................................. 55
Unit 1 Section 4 Post-Test ............................................................... 56
Unit 1 Section 5: Data Types .......................................................... 57
  Purpose: .......................................................................................... 57
  Overview: ....................................................................................... 57
  Problem Introduction: ................................................................. 58
  Problem Programming Steps: ..................................................... 58
  Individual Problems: ................................................................... 60
  Tips: ............................................................................................... 60
  For More Information: ................................................................. 60
Unit 1 Section 5 Pre-Test ................................................................. 61
Unit 1 Section 5 Post-Test ............................................................... 62
Unit 1 Section 6: Concatenation ..................................................... 63
  Purpose: .......................................................................................... 63
  Overview: ....................................................................................... 63
  Problem Introduction: ................................................................. 64
  Problem Programming Steps: ..................................................... 64
Unit 2 Section 6 .................................................................................................................. 204
Unit 2 Section 7 .................................................................................................................. 204
Unit 2 Section 8 .................................................................................................................. 204
Unit 2 Section 9 .................................................................................................................. 204
Unit 2 Section 10 (Unit 2 Review) ..................................................................................... 204
Unit 3 Section 1 .................................................................................................................. 204
Lesson Overview

In this section, the purpose of each lesson will be presented.

Unit 1: Basics

- **Section 1: Layout of a Webpage**: The purpose of this lesson is to explain the basics of computer programming. The simple pages that you will create in this lesson will consist of text.
- **Section 2: Buttons**: The purpose of this lesson is to introduce you to buttons.
- **Section 3: Simple Logic**: The purpose of this lesson is to explain how to use simple logic. We will teach you how to use if statements and else statements.
- **Section 4: Textboxes and Tables**: The purpose of this lesson is to teach you how to use textboxes and tables. Therefore, the websites that you can create will become more dynamic, interactive, and organized.
- **Section 5: Data Types**: The purpose of this lesson is to teach you about different data types. You will be reintroduced to strings. You will be introduced to decimals, strings, and the Boolean data type. Also, you will learn how to convert between data types.
- **Section 6: Concatenation**: The purpose of this lesson is to learn concatenation. By learning concatenation, you can make more complex websites.
- **Section 7: Multi-Page Websites**: The purpose of this lesson is to teach you how to link multiple pages into a website. Also, color changes will be introduced.
- **Section 8: Variables**: The purpose of this lesson is to teach you about variables.
- **Section 9: Radio Button Lists, Check Box Lists, and Drop Down Lists**: The purpose of this lesson is to introduce you to three more objects: Radio Button Lists, Check Box Lists, and Dropdown Lists.
- **Section 10: Loops**: The purpose of this lesson is to teach you about loops. Loops run a segment of code over and over again until they are told to stop. Also, information about counters and operators is included.

Unit 2: Database Operations

- **Section 1: SQL Select 1**: The purpose of this lesson is to introduce you to SQL.
- **Section 2: SQL Select 2**: In Unit 2 Section 1, you learned about databases, and how to display data in a database. The purpose of this lesson (Unit 2 Section 2) is to teach you how to display data in a web application.
- **Section 3: SQL Select 3**: The purpose of this lesson is to introduce you to advanced select statements.
- **Section 4: Logins**: The purpose of this lesson is to teach you how to make a database-based login. Also, session variables and documentation will be introduced.
- **Section 5: SQL Select 4**: The purpose of this lesson is to teach you how to create search functions for gridviews in your application.
- **Section 6: SQL Insert**: The purpose of this lesson is to teach you how to insert records into the database. First, you will use insert statements within the database, then you will create a web application in which the user can add records to the database.
- **Section 7: SQL Select 5**: The purpose of this lesson is to teach you how to select data from a gridview so that the user can manipulate it using objects.
- **Section 8: SQL Delete**: The purpose of this lesson is to teach you how to delete a record from a table using SQL.
- **Section 9: SQL Update**: The purpose of this lesson is to teach you how to update records using SQL.
- **Section 10: SQL Combination (Unit 2 Review)**: The purpose of this lesson is for you to review Unit 2. You will combine all of the knowledge you have gained from this Unit (SQL Select, SQL Insert, SQL Update, SQL Delete, and Logins).

**Unit 3: Other**
- **Section 1: Arduino Stoplight Simulator**: The purpose of this lesson is to demonstrate the capabilities of the Arduino microcontroller.
Topic List

- Layout (1.1)
  - Script (1.1)
  - Body (1.1)
  - Create a page without code behind (1.1)
    - Source Mode (1.1)
    - Design Mode (1.1)
    - Split Mode (1.1)
  - Multi-Page Websites (1.7)
  - Background color of Document (1.7)
- Logic
  - If then statements (1.3)
  - If then else statements (1.3)
  - Nested if statements (1.4)
  - Loops (1.10)
    - Do/loop while (1.10)
    - Do/Loop until (1.10)
    - For/next (1.10)
  - Operators
    - And (1.4)
    - & (concatenation) (1.6)
    - Addition/subtraction/multiplication/division (1.5)
    - += (1.9)
    - = (1.1)
    - <> or != (not equal) (1.10)
- Objects
  - Textbox (1.4)
  - Buttons (1.2)
  - Label (1.1)
  - Radio buttons (1.9)
  - Check boxes (1.9)
  - Drop Down Lists (1.9)
  - Organization-tables (1.4)
  - Menu (1.7)
    - Auto Format (1.7)
- Sub Routines
  - Sub Page load (1.1)
  - Button_Click (1.2)
- Gridview_SelectedIndexChanged (2.7)

**Attributes/Properties**
- General (1.3)
- ID (1.1)
- Text (1.1)
- Visible (1.2)
- Selected Value (1.9)

**Variables (1.8)**
- String (1.8)
- Integer (1.8)
- Decimal (1.8)
- Boolean (1.8)
- Session Variable (2.4)

**Database Operations**
- MySQL Workbench (2.1)
- Creating a DSN (2.2)
- What a database looks like (2.1)
  - Table (2.1)
  - Row (2.1)
  - Column (2.1)
  - Schema (2.1)
  - Primary key (2.2)
- Import namespace (2.2)
- Login (2.4)
- Grid View (2.2)
  - Databind (2.5)
  - Enable Paging and Sorting (2.3)
  - Enable Selection (2.7)
  - Selected Row (2.7)
- SQL DataSource (2.2)
  - SQL (2.1)
    - Select (2.1, 2.2, 2.3, 2.5, 2.7)
    - Insert (2.6)
    - Update (2.9)
    - Delete (2.8)
    - Combination (2.10)

**Data Types (1.5)**
- String (1.5)
- Integer (1.5)
- Decimal (1.5)
- Boolean (1.5)
- Conversion of Data Types (1.5)

❖ HTML
- `<html> </html>` (1.1)
- `<head> </head>` (1.1)
- `<title> </title>` (1.1)
- `<body> </body>` (1.1)
- `<form> </form>` (1.1)
- `<br />` (1.3)
- `<p> </p>` (1.1)
- `<h1> </h1>` (1.1)
- `<div> </div>` (1.1)
- `<a> </a>` (1.7)

❖ Other
- Visual Web Developer Introduction (1.1)
- How to Debug an application (1.1)
- Arduino (3.1)
- Opening the Toolbox (1.1)
- Object Property/Attribute Introduction (1.1)
- Comments (1.4)
- Response.Write (1.1)
- Response.Redirect (1.7)
- Counters (1.10)
- Random Numbers (1.10)
- Documentation (2.4)
Downloads For This Course
Author: Hadiya
Date Created: 04-26-14

Purpose:
The purpose of this information page is for the student to download the needed software for this course. The necessary downloads include Visual Web Developer, MySQL Community Server, MySQL Workbench, and the MySQL ODBC Driver.

For each, choose the 32-bit MSI installer. If you see a choose platform box, select whichever operating system that you have (ex. Mac OS X, Microsoft Windows).

**Downloading Visual Web Developer:**
Use the following link:

http://www.asp.net/vwd

**Downloading MySQL Workbench:**
Use the following link:

http://dev.mysql.com/downloads/tools/workbench/

**Downloading MySQL Community Server:**
Use the following link:

http://dev.mysql.com/downloads/mysql/5.5.html

**Tips:**
- Keep clicking next until you get to the end of the setup.
- When prompted, choose a developer machine as a server type and multifunctional database.
- Make sure that you remember the password that you create during setup!
  - The username for this password is root.

**Downloading MySQL ODBC Driver:**
Use the following link:

https://dev.mysql.com/downloads/connector/odbc/

Download the **5.1.13 Connector**!!!

1. Click ‘Looking for previous GA versions?’.
2. Select 5.1.13
Unit 1 Section 1: Layout of a Web Page
Authors: Kiese, Hadiya, Hasani
Date Created: 3-1-2014

Please take the Pre-Test before the start of this lesson.

Purpose: The purpose of this lesson is to explain the basics of computer programming. The simple pages that you will create in this lesson will consist of text.

Overview:
Please read http://www.w3schools.com/html/html_intro.asp for an introduction to html. HTML stands for Hyper Text Markup Language, and is the basic structure for many webpages. Read from the beginning of the page to the HTML Versions section. Then, play around with the Try It Yourself Example at the top by clicking the “Try it yourself” button.

HTML Introduction

You can change the text between the <h1> tags and the <p> tags.
Click “Submit Code” to see the results of your code:

Also, you can add more paragraph tags and h1 tags:

*Remember to have both beginning and ending tags (ex. <p> insert text here </p> OR <h1> insert text here </h1>). There should be a slash in the end tag (ex. </h1> or </p>).*

Secondly, read [http://www.w3schools.com/html/html_editors.asp](http://www.w3schools.com/html/html_editors.asp) to learn about HTML Editors. One HTML Editor that you can use is Notepad. Open Notepad and follow the directions given on the website. You may not have Notepad if you have an Apple Computer. If so, skip this step.

Now that you have created a web page using notepad, we will introduce you to a more advanced html editor called Visual Web Developer. Visual Web Developer uses ASP.Net.
Please read the description of ASP.net in the picture below from http://www.asp.net/get-started and the associated definitions.

**Web Sites**

ASP.NET is great for building standards-based websites with HTML5, CSS3, and JavaScript. ASP.NET supports three approaches for making web sites. ASP.NET Web Forms uses controls and an event-model for component-based development. ASP.NET MVC values separation of concerns and enables easier test-driven development. ASP.NET Web Pages prefers a single page model that mixes code and HTML markup. You can mix and match these techniques within one application depending on your needs - it's all One ASP.NET.

Definitions:
- CSS - Cascading Style Sheets (the colors and layout of webpages)
- JavaScript - can be used to “program the behavior of web pages” (see http://www.w3schools.com/js/DEFAULT.asp)
- ASP.Net MVC - Model View Controller (see http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework for more information)

**Visual Web Developer Instructions:**
1. Open Visual Web Developer (Click start and find Visual Web Developer). The icon appears in a picture below

2. In the top left corner of the screen, click new website.
3. In the new window, change “WebSite1” to name your website. Do not change any of the other information in this text box. We suggest that you name this website HelloWorld (without the space in between Hello and World). Make sure that “ASP.Net Website” is selected. Then click, “OK” to create your website.

4. First, we need to make a page without code behind. Code behind puts the background code of a page into a separate page. If a page is created without code behind, then the same information is put into the script section of the code at the top of the page. To make a page without code behind, first click Website and go to Add New Item.
5. Next, make sure “Web Form” is selected. Then, change the name “Default2”, making sure that your name still ends in “.aspx”. We recommend that you change the name to “helloworld1”. Lastly, and most importantly, uncheck the box “place code in a separate file” which creates a code behind file.

6. After you click “Add”, a page like this will appear.
7. This page is currently in “Source” mode. The “Source” version of a webpage allows you to see the code that creates your webpage. You can tell that you are in “Source” mode by looking at the bottom of the screen.

In the source mode, there are two main parts of the code: the script and the html sections. Each has a starting tag (<script> and <html>) and an ending tag (<script> and </html>). The other parts of the tags (i.e. runat="server" and xmlns="http://www.w3.org/1999/xhtml") describe their corresponding section (script and html).

8. In the html section, there are also two sections: head and body. The head section has starting and ending tags (<head> and </head> respectively) contains background information on the webpage such as the “title”. The title section also has starting and ending tags (<title> and </title> respectively). The “body” section will contain other html tags that you have used previously and objects that you will be introduced to shortly.
9. The form tags serve as a container for objects, html tags, and text. As always there is a starting and ending form tag (<form> and </form> respectively). Lastly, the div tag creates a section of the page in which to put other tags, objects, and text. Once again, there is a starting and ending tag for this section (<div> and </div> respectively).

```
<form id="form1" runat="server">
  <div>
  </div>
</form>
```

10. Next, change the webpage into “Design” mode by clicking “Design” at the bottom of the screen.

11. You will now see a page like this:
    There is a virtual representation of the “div” that you just saw in the code in #9. This is what the code in the “Source” section programs the webpage to look like.

12. Next, click the “Split” mode.

13. The “Split” mode displays a combination of the “Design” and “Source” modes.
Problem Introduction:
There are many ways to display text using ASP.Net. The problems below will teach you four ways of displaying “Hello World”

Problem 1: Create a webpage that displays “Hello World” by typing in a div.
Problem 2: Create a webpage that displays “Hello World” by using the <h1> or heading 1 tag.
Problem 3: Create a webpage that displays “Hello World” by using the <p> or paragraph tag.
Problem 4: Create a webpage that displays “Hello World” by using an asp.net label.
Problem 5: Create a webpage that displays “Hello World” using Response.write.
Problem 6: Create a webpage that displays your name using any method.
Problem 7: Create another webpage that displays your name using another method.

Problem Programming Steps:
Problem 1: Create a webpage that displays “Hello World” by typing in a div.

14. The simplest way to place text on the screen is by simply typing. Problem 1 asks us to type in the div. First, choose a mode (Source, Split, or Design).
   a. If you go to “Source” mode, type Hello World between the start and ending div tags. Then go to “Design” mode to see Hello World on your webpage.

   ```
   <script runat="server">
   </script>
   <html xmlns="http://www.w3.org/1999/xhtml">
   <head runat="server">
   <title></title>
   </head>
   <body>
   <form id="form1" runat="server">
   <div>
   Hello World
   </div>
   </form>
   </body>
   </html>
   ```

   b. If you go to “Design” mode, type Hello World in the div.

   ![div](Image)

   c. If you go to “Split mode” use either method in 14a or 14b. You may need to refresh the design mode part of the page by clicking the banner that appears above it if prompted.
15. Regardless of any mode that you used in the previous step, to see the web page in a web browser, go to the “Debugging” menu and click “Start Debugging.”

16. If you get the following error, select “Modify the Web.config file to enable debugging.” then click “OK”.

17. Next, a webpage should open in your default internet browser that says Hello World.
18. To stop debugging, click the red x in your web browser to close it. Go back to Visual Web Developer. If the application is still “(Running)”, then go to the “Debug” menu, and click “Stop Debugging”.

19. For the next problem create a new page (see steps 4, 5, and 6). We recommend that you name this page helloworld2.

20. Then using what you learned from the Overview section, display Hello World by using the heading 1 tag. The heading 1 tag can appear anywhere in between the body tags.

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Hello World</title>
</head>
<body>
  <h1>Hello World</h1>
</body>
</html>
```

21. Lastly, debug your code to view it as a webpage. See step 15 for more help.

**Problem 3: Create a webpage that displays “Hello World” by using the <p> or paragraph tag.**

22. Problem 3 is a variation of problem 2. Make sure that you create a new page and debug it when you are done. Do not forget to include both the beginning and ending paragraph tags.
Problem 4: Create a webpage that displays “Hello World” by using an asp.net label.

23. First, create a new webpage (See steps 4-6).
24. Secondly, we need to get a label. A label is an object. We are currently using an object oriented programming language. A definition of an object oriented programming language from Wikipedia is below:

Object-oriented programming (OOP) is a programming paradigm that represents concepts as "objects" that have data fields (attributes that describe the object) and associated procedures known as methods. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.


Programming paradigm: style of programming

Class: defined by Wikipedia as: a template for creating objects


25. Because a label is an object, it will be found in your toolbox. First, switch to “Design” mode. The toolbox is located near the top left of your screen. Click the toolbox icon.
26. From here a box will open up with a variety of objects in it. Select the “Label” and drag it into your webpage.

27. You will see the label appear in the design view.
28. Now, while the label is still selected, look at the bottom right corner. You will see a properties section. If you do not see Label1 in the drop down list, then select it from the drop down list. The table in this section displays properties and attributes of the label. The “Text” of the label is “Label”. To change the “Text” property of the label, simply click on “Label” and change the text to Hello World. Note the description of the Text property of the label below the properties section.

29. Next, you need to name the label. Name the label by scrolling through the properties of the label until you find “ID”. Change the “ID” to any name that you want without using spaces. We suggest that you name the label lblhelloworld. Using the “lbl” prefix before your name will be very helpful when programming in the future so that you can determine which type of object you are using from the ID of the object.
30. Please notice that the changes that you have made in the “Design” mode are reflected in the “Source” mode. In the “Source” mode, the format of these changes are `<tagname property="attribute"> </tagname>`. More properties can be added after the first property. Therefore, for your label, ID="lblhelloworld" and Text="Hello World".

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="lblhelloworld" runat="server" Text="Hello World"></asp:Label>
    </div>
  </form>
</body>
</html>
```

31. Debug to test your label.

**Problem 5: Create a webpage that displays “Hello World” using Response.write.**

32. Another way to write Hello World is to use the script. The script executes each line of code in order. As usual, the script has both a starting and ending tag. The script is located at the top of the page in the “Design” mode.

First, create a new page. Make sure that this page does not have code behind/does not have a separate code file.

33. Next, go to the “Design” view and double click in the open space. Make sure that you do not double click in the div that is on the page.

```
Double click somewhere out here
```
34. You have now created a subroutine or “Sub” in the script tag of your code that will execute right after the page loads. Therefore, the Sub is called “Sub Page_Load”. As usual, this subroutine has both starting and ending tags (Sub and End Sub respectively). Any code that is placed in between these tags will run upon page load.

35. To write text in the script, we use the command Response.Write(“Text”). To write hello world, replace “Text” with “Hello World”. Therefore, the command is Response.Write(“Hello World”) Make sure that you include the quotation marks on both sides. These quotation marks signal that the words inside the quotation marks are text that is meant to be displayed.

36. Notice that if you switch back to “Design” view, Hello World does not appear. This is because the web page will write Hello World onto the screen when you actually load the webpage and Sub Page_Load runs. Sub Page_Load has not run yet and will not run until you debug.

37. Lastly, debug the website to view your webpage in a web browser.

Individual Problems:
Problems 6 and 7 are both individual problems. If you need help, see the Tips Section (below), Overview Section (above), and Problem Programming Steps Section for help.

Problem 6: Create a webpage that displays your name using any method.

Problem 7: Create another webpage that displays your name using another method.

Tips:
• Naming is very important. Be sure to name every object, website, etc. appropriately.

For More Information:
• http://www.w3schools.com/html/default.asp
• http://www.w3schools.com/tags/default.asp
Unit 1 Section 1 Pre-Test

Please take this test before the lesson.

Name:___________________________________

Directions: Please explain what each line of code does by answering the multiple choice question.

1. Response.Write(“My name is Bob”)
   a. Respond to Bob’s email
   b. Write My name is Bob on the screen
   c. Insert Bob into a database table called Response.Write
   d. Tell Bob that his name is Bob

2. &lt;asp:Label ID="name" runat="server" text="My name is Amy"&gt; &lt;/asp:Label&gt;
   a. Put Amy on the server
   b. Label Amy’s name on the server
   c. Put My name is Amy into a Label
   d. Define My name is Amy as a name

3. &lt;p&gt;My name is Bob&lt;/p&gt;
   a. Put My name is Bob in a paragraph
   b. Parse My name is Bob
   c. Define Bob as a peripheral
   d. Make Bob the president

4. &lt;h1&gt;My name is Amy&lt;/h1&gt;
   a. Hide Amy’s record
   b. Delete Amy’s record from the database
   c. Help Amy
   d. Put My name is Amy into a heading

5. &lt;div&gt;My name is Bob&lt;/div&gt;
   a. Put My name is Bob into a section called a div
   b. Divide My name is Bob into characters
   c. Divide My name is Bob into words
   d. Descriptively Insert a Variable called My name is Bob
Unit 1 Section 1 Post-Test
Please take this test after the lesson. [Answers](#)

Name: ________________________________

Directions: Please explain what each line of code does by answering the multiple choice question.

1. Response.Write(“My name is Bob”)
   a. Respond to Bob’s email
   b. Write My name is Bob on the screen
   c. Insert Bob into a database table called Response.Write
   d. Tell Bob that his name is Bob

2. `<asp:Label ID=”name” runat=”server” text=”My name is Amy”> </asp:Label>`
   a. Put Amy on the server
   b. Label Amy’s name on the server
   c. Put My name is Amy into a Label
   d. Define My name is Amy as a name

3. `<p>My name is Bob</p>`
   a. Put My name is Bob in a paragraph
   b. Parse My name is Bob
   c. Define Bob as a peripheral
   d. Make Bob the president

4. `<h1>My name is Amy</h1>`
   a. Hide Amy’s record
   b. Delete Amy’s record from the database
   c. Help Amy
   d. Put My name is Amy into a heading

5. `<div>My name is Bob</div>`
   a. Put My name is Bob into a section called a div
   b. Divide My name is Bob into characters
   c. Divide My name is Bob into words
   d. Descriptively Insert a Variable called My name is Bob
Unit 1 Section 2: Buttons

Author: Hadiya
Date Created: 3-22-2014

Please take the Pre-Test before the start of this lesson.

Purpose: The purpose of this lesson is to introduce you to buttons.

Overview:

Buttons:
A button is an object. When you click on the button, it will perform a subroutine. By editing the properties of the button and the subroutines that it performs, you can create an interactive website. See the problem programming steps for instructions for how to create a button.

Referring to objects:
- To refer to an object and its properties, use the following form:
  
  `ObjectName.Property`

  For example:
  - The Text of a Button that is called Submit is referred to as Submit.Text
  - The ID of a Button that is called Submit is referred to as Submit.ID
  - The Visibility of a Button that is referred to as Submit.Visible

- When setting an object’s properties equal to something, there are many forms.
  - To set Submit.Text to Go Back use Submit.Text="Go Back"
    - The text is in double quotes because it is composed of words/text that will be printed out as words/text. This helps to distinguish from a variable, which doesn’t use double quotes.
  - To make the Submit button visible use Submit.Visible=True
    - The True value is not in double quotes because there is no text that will be printed out that says true. The same works for False. True and False are Boolean values and not strings.

Problem Introduction:

**Problem 1:** Create a button that makes ‘Hello World’ visible. Create another button that makes ‘Hello World’ invisible.

**Problem 2:** Create a button that makes your name visible. Create another button that makes your name invisible.
Problem Programming Steps:

Problem 1: Create a button that makes ‘Hello World’ visible. Create another button that makes ‘Hello World’ invisible.

1. Create a new website. Make sure that you name your website. We suggest that you name your website ‘visibleinvisible’. See Unit 1 Section 1 if you need help.
2. Create a new page without code behind. We suggest that you name your page ‘helloworld’ See Unit 1 Section 1 if you need help.
3. Drag in a label and two buttons from the toolbox. Name the label and textbox. We suggest that you name the label ‘lblhelloworld’ and the buttons ‘btnvisible’ and ‘btninvisible’.
4. Put the text ‘Hello World’ into the label. Also, the text of ‘btninvisible’ should be ‘Invisible’. The text of ‘btnvisible’ should be ‘Visible’

This looks very disorganized. One method of organizing this is to use paragraphs. In this example, we will use two paragraphs to organize the buttons.

First, switch to ‘Split’ or ‘Source’ mode. Then, add paragraph tags (<p> </p>) around the label and around the two buttons. Your code should now look like this:
The result in ‘Design’ mode is this:

![Design Mode Image]

You can click between the buttons and add a few spaces if you would like to (just as if you were separating words in a word processing program).

Spaces appear in the source code as ‘&nbsp;’.

6. Next, we will tell the buttons what to do when they are clicked. Let’s start with the ‘Visible’ button. Double click the button in ‘Design’ mode to create a subroutine in ‘Source’ mode.

**Design Mode:**

![Design Mode Image]

**Source Mode:**

```csharp
<%Protected Sub btnvisible_Click(sender As Object, e As System.EventArgs)%>
    lblhelloworld.visible=True
</%>
```

7. In the Subroutine, we need to tell the button to make the label ‘lblhelloworld’ invisible. A label is an object that has many properties. For example, the ID is the name of the variable; the text is the text displayed in the label. Another property is visible. Visible tells the website whether the object (in this case, a label) is visible or invisible.

If the label is visible, then the property for visible is true. This is written in this form:

```csharp
Propertiname.visible=True
```

If the label is invisible, then the property for visible is false. This is written in this form:

```csharp
Propertiname.visible=False
```
Therefore, in the subroutine for the ‘Visible’ button, we need to write:
‘lblhelloworld.Visible=True’

```vbnet
Protected Sub btnvisible_Click(sender As Object, e As System.EventArgs)
    lblhelloworld.Visible = True
End Sub
```

8. Next, we will do the invisible button. Go to ‘Design’ mode and double click the ‘Invisible’ button.
In the newly created subroutine, type the code to ‘lblhelloworld.visible=False’

```vbnet
Protected Sub btninvisible_Click(sender As Object, e As System.EventArgs)
    lblhelloworld.Visible = False
End Sub
```

9. Finally, you should end up with the code below. You are now ready to debug your application to test it! See Unit 1 Lesson 1 if you do not remember how to debug.

---

**Individual Problem:**

**Problem 2:** Create a button that makes your name visible. Create another button that makes your name invisible. Do this problem on your own. However, we have provided some Tips below to help you:

**Tips:**

- Name all of your objects, pages, and websites wisely. We suggest that you start the name of all buttons with ‘btn’, all labels with ‘lbl’, and all textboxes with ‘txt’. If you use these naming conventions, then while you type, the intellisense that Visual Web Developer uses will easily
recognize that you are searching for which ever object that you are looking for. Also, you will know what type of object you have by simply looking at the name of an object.

- If you are unsure of the properties of an object, click the object and look at the properties panel on the bottom right.

For more information:

Unit 1 Section 2 Pre-Test

Please take this test before the lesson.

Name: _______________________________________

Directions: Please explain what each line of code does by selecting a multiple choice answer choice:

1. btnHelp.visible=False
   a. Make btnHelp invisible
   b. Make btnHelp visible
   c. Change the text of btnHelp to False
   d. This line of code would produce an error

   a. Put the help label in parenthesis
   b. Display help in a label. The label is contained in a paragraph.
   c. Display Hello World in a label. The label is contained in a paragraph.
   d. Display Hello World in a label. The label is contained in parenthesis.

Directions: Tell how the following line of code will appear in the ‘Design’ mode by selecting a multiple choice answer option:

1. <p> <asp:Label ID="help" runat="server" text="Hello World"/> </asp:Label> </p>
   <p> <asp:Label ID="help1" runat="server" text="Hello Friend"/> </asp:Label> </p>
   <p> <asp:Button ID="help2" runat="server" text="Hello"/> </asp:Label> </p>

   a. Option 1:  
   b. Option 2:  
   c. Option 3:  
   d. Option 4:
Unit 1 Section 2 Post-Test
Please take this test after the lesson. Answers

Name: _____________________________________

Directions: Please explain what each line of code does by selecting a multiple choice answer choice:

2. btnHelp.visible=False
   e. Make btnHelp invisible
   f. Make btnHelp visible
   g. Change the text of btnHelp to False
   h. This line of code would produce an error

3. <p><asp:Label ID="help" runat="server" text="Hello World"> </asp:Label> </p>
   a. Put the help label in parenthesis
   b. Display help in a label. The label is contained in a paragraph.
   c. Display Hello World in a label. The label is contained in a paragraph.
   d. Display Hello World in a label. The label is contained in parenthesis.

Directions: Tell how the following line of code will appear in the ‘Design’ mode by selecting a multiple choice answer option:

1. <p><asp:Label ID="help" runat="server" text="Hello World"> </asp:Label> </p>
   <p><asp:Label ID="help1" runat="server" text="Hello Friend"> </asp:Label> </p>
   <p><asp:Button ID="help2" runat="server" text="Hello"> </asp:Label> </p>

c. Option 1: c. Option 3: 

   d. Option 2: d. Option 4: 
Unit 1 Section 3: Simple Logic
Author: Hadiya
Date Created: 3-22-2014

Please take the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to explain how to use simple logic. We will teach you how to use if statements and else statements.

Overview: If (insert a condition here) then (do this) is a common pattern used while speaking. For example: If the meat is on sale then buy some for me. Also, if I see a dangerous object, then I will run away from the situation. These words (“If” and “then”) also carry over to programming. For example: If the user types hello then change the text to goodbye. We can also add the word else. For example: If the meat is on sale, then buy some for me, else wait until next week. Also, if I see a dangerous object, then I will run away from the situation, else I will go to the grocery store. The word else is also used in programming. For example: If the user types hello then change the text to goodbye, else change the text to hello world.

Problem Introduction:
Problem 1: If the label says hello then make it say goodbye using a button. If the label says goodbye then make it say hello. Use only one button to accomplish this task.

Problem 2: A label displays ‘Hello World’. If the label is visible then make it invisible. If the label is invisible then make it visible. Use only one button to accomplish this task.

Problem Programming Steps:
Problem 1: If the label says hello then make it say goodbye using a button. If the label says goodbye then make it say hello. Use only one button to accomplish this task.

1. Create a new website. We recommend that you call this website ‘ifstatements’. Please see Unit 1 Section 1 if you need help.
2. Create a new page without code behind. We recommend that you call this page ‘hellogoodbye.aspx’. Please see Unit 1 Section 1 if you need help.
3. Put a label and a button on the page. We recommend that you name the label ‘lblhellogoodbye’ and the button ‘btntoggle’.
5. Next, we will organize the page. To do this we will use the html break tag. This tag is similar to pressing the enter button. Type <br /> in between the label and the button while in ‘Source’ mode to achieve this effect.

6. Now, we will create an if statement that tells the button what to do in a certain scenario. The format of the if statement is as follows:

   If *insert condition*
   Then *do something*

Because we want the label to toggle the text between hello and goodbye, we must create an if statement to tell the button what to do. The first step is to change the text from Hello to Goodbye. In plain English: If the label says Hello, change the text to Goodbye. The result is the if statement below:

```
If *insert condition*
Then *do something*
```

Notice that we use the line lblhellogoodbye.Text often. This refers to the name of the label and the property that we want to change. The name of the label or ID is lblhellogoodbye. The property that we want to work with is the Text of the label.

Also notice that the words that we want to change the text to are in double quotes. This is because “Hello” and “Goodbye” are strings. A string is composed of text. Therefore, for all properties where text is involved, the text must be put in double quotes.
7. Try debugging your application. When you click the button, it should change from Hello to Goodbye. However, it will not change back to Hello because we have not programmed the button to do this. We can do this by adding an else statement. In plain English: If the text says Hello, then change it to Goodbye, else change the text to Hello. This also works the other way: If the text says Goodbye, then change the text to Hello, else change the text to Goodbye.

The code for the first case is below:

```xml
<asp:script runat="server">

    Protected Sub btnToggle_Click(sender As Object, e As System.EventArgs)
    If lblHelloGoodbye.Text = "Hello" Then
        lblHelloGoodbye.Text = "Goodbye"
    Else
        lblHelloGoodbye.Text = "Hello"
    End If

End Sub
</asp:script>
```

8. Debug your application. Now try changing the code the other way so that if the text says Goodbye, then it will be changed to text to Hello, else change the text to Goodbye.

**Problem 2:** A label displays ‘Hello World’. If the label is visible then make it invisible. If the label is invisible then make it visible. Use only one button to accomplish this task.

9. Make a new page without code behind. We recommend that you call this page ‘visibleinvisible.aspx’. Make sure that you read the tips below before proceeding with this problem.

**Tips:**
- To refer to an object and its properties, use the following form:

  ```
  ObjectName.Property
  ```

  For example:
  - The Text of a Button that is called Submit is referred to as Submit.Text
  - The ID of a Button that is called Submit is referred to as Submit.ID
  - The Visibility of a Button that is referred to as Submit.Visible

- When setting an object’s properties equal to something, there are many forms.
  - To set Submit.Text to Go Back use Submit.Text="Go Back"
    - The text is in double quotes because in is composed of words/text that will be printed out as words/text. This helps to distinguish from a variable, which doesn’t use double quotes.
  - To make the Submit button visible use Submit.Visible=True
The True value is not in double quotes because there is no text that will be printed out that says true. The same works for False. True and False are Boolean values and not strings.

For More Information:
Unit 1 Section 3 Pre-Test

Please take this test before the lesson.

Name: ____________________________

Directions: Please explain the outcome of each section of code. Read through each example in its entirety. What will happen when I click the button?

1. 
   ```html
   Hello will change to Goodbye
   Daytime will change to Nightime
   Nightime will change to Daytime
   Nothing will change
   Error
   ```

2. 
   ```html
   Hello will change to Goodbye
   Daytime will change to Nightime
   Nightime will change to Daytime
   Nothing will change
   Error
   ```

3. 
   ```html
   Hello will change to Goodbye
   Daytime will change to Nightime
   Nightime will change to Daytime
   Nothing will change
   Error
   ```
Unit 1 Section 3 Post-Test

Please take this test after the lesson. Answers

Name: ____________________________

Directions: Please explain the outcome of each section of code. Read through each example in its entirety. What will happen when I click the button?

1.

a. Hello will change to Goodbye
   b. Daytime will change to Nightime
   c. Nightime will change to Daytime
   d. Nothing will change
   e. Error

2.

a. Hello will change to Goodbye
   b. Daytime will change to Nightime
   c. Nightime will change to Daytime
   d. Nothing will change
   e. Error

3.

a. Hello will change to Goodbye
   b. Daytime will change to Nightime
   c. Nightime will change to Daytime
   d. Nothing will change
   e. Error
Unit 1 Section 4: Textboxes and Tables
Author: Hadiya
Date Created: 3-29-2014

Please take the Pre-Test before beginning this lesson.

Purpose:
The purpose of this lesson is to teach you how to use textboxes and tables. Therefore, the websites that you can create will become more dynamic, interactive, and organized.

Overview:
A text box is very similar to a label. For example, both of these objects have a text property. However, the user is able to directly type into the text box to enter information. By including text boxes in your website, the site becomes more interactive.

The layout of a website is very important. One way to organize a website effectively is to use a table. You can put objects such as a textbox, label, button and more into a table.

Problem Introduction:
Problem 1: Create a typing teacher. The user will see 5 textboxes and will be prompted to type one word per textbox. The words are: cat, dog, bear, yellow, and blue. After the user clicks the Check button, tell the user whether or not the words were typed correctly.

Problem 2: Create a login system. Use a table to organize the login system. The username is ‘HelloWorld’ (without the tick marks), and the password is ‘12345’ (without the tick marks). If the username and password are correct display a label that says User is Authenticated. If the username and password are not correct then display a label that says the User is Not Authenticated.

Problem Programming Steps:
Problem 1: Create a typing teacher. The user will see 5 textboxes and will be prompted to type one word per textbox. The words are: cat, dog, bear, yellow, and blue. After the user clicks the Check button, tell the user whether or not the words were typed correctly.

1. Create a new website. We recommend that you call the website ‘textboxesandtables’. Create a new page without code behind. We recommend that you call the page ‘typingteacher.aspx’.
2. Next we will create a table. Go to the top menu, then select Table, Insert Table.
3. Create a table that has 7 rows and 3 columns. Click ‘OK’ to create the table.

4. Drag the appropriate objects into the table as shown below. You can also resize the table as needed.
5. Next, we will name everything and insert the correct text. We recommend that the first Label should be used as a Title. Therefore, we will name it ‘lbltitle’ and the text will be ‘Typing Teacher’.

6. The rest of the labels in the first column are for the 5 words that we ask the student to type. Therefore, I will name them chronologically ‘lblcat’, ‘lbldog’, ‘lbldbear’, ‘lblyellow’, and ‘lblblue’. The text will be chronologically ‘1) cat’, ‘2) dog’, ‘3) bear’, ‘4) yellow’, and ‘5) blue’.

7. Name the button ‘btncheck’. The text of the button should be ‘Check my answers’.


9. Name the labels in the third column ‘lblcheckcat’ ‘lblcheckdog’ ‘lblcheckbear’ ‘lblcheckyellow’, and ‘lblcheckblue’. The text property should be empty. Your ‘Design’ mode screen should appear like the image below:

![Typing Teacher interface](image-url)
10. Next, double click the check button to create a subroutine for when the check button is clicked. In order to check if the user typed each word correctly, use if statements. If the user types the word correctly, display a correct message in the corresponding label in the third column. If the user types the word incorrectly, display an incorrect message in the corresponding label in the third column; this can be done using an else statement. If you need help, the code for this is below:

```<script runat="server">
Protected Sub btncheck_Click(sender As Object, e As System.EventArgs)
    'cat
    If txtcat.Text = "cat" Then
        lblcheckcat.Text = "Correct"
    Else
        lblcheckcat.Text = "Incorrect. Please try again."
    End If
    
    'dog
    If txtdog.Text = "dog" Then
        lblcheckdog.Text = "Correct"
    Else
        lblcheckdog.Text = "Incorrect. Please try again."
    End If
    
    'bear
    If txtbear.Text = "bear" Then
        lblcheckbear.Text = "Correct"
    Else
        lblcheckbear.Text = "Incorrect. Please try again."
    End If
    
    'yellow
    If txtyellow.Text = "yellow" Then
        lblcheckyellow.Text = "Correct"
    Else
        lblcheckyellow.Text = "Incorrect. Please try again."
    End If
    
    'blue
    If txtblue.Text = "blue" Then
        lblcheckblue.Text = "Correct"
    Else
        lblcheckblue.Text = "Incorrect. Please try again."
    End If
End Sub
</script>
```

Notice that there are green lines, which do not contain code. These lines start with tick marks ('). They are called comments. Comments do not run as code; they are notes that make the code that you write easier to read. We used comments in this example to divide each section of if/then/else statements. We recommend that you use comments to make your code more understandable.

11. Next, debug your application and try it. Make sure that both the correct and incorrect functions work.
Problem 2: Create a login system. Use a table to organize the login system. The username is HelloWorld the password is 12345. If the username and password are correct display a label that says User is Authenticated. If the username and password are not correct then display a label that says the User is Not Authenticated.

12. For this problem, create a new page without code behind and work on the problem on your own. Please make sure that you read the Tips section below before starting.

Tips:

- Nested if Statements; And Operator:
  - If statements can be nested within one another. For example, if I wanted to check if all of the textboxes in the previous section were correct and display a message in lblcheckcat if all of the text boxes were filled in correctly, I can use the code below:

```vbnet
Protected Sub btncheck_Click(sender As Object, e As System.EventArgs)
    If txtcat.Text = "cat" Then
        If txtdog.Text = "dog" Then
            If txtbear.Text = "bear" Then
                If txtyellow.Text = "yellow" Then
                    If txtblue.Text = "blue" Then
                        lblcheckcat.Text = "All entries are correct"
                    End If
                End If
            End If
        End If
    End If
End Sub
```

  - I can also do this by using the operator ‘and’:

```vbnet
Protected Sub btncheck_Click(sender As Object, e As System.EventArgs)
        lblcheckcat.Text = "All entries are correct"
    End If
End Sub
```

- Comments:
  - Use comments often. They are very helpful so that you do not lose track of what you are doing.
  - One technique for using comments is to make a list of things to do using comments, then type the code in between. For example:
- ‘check if password is 8 characters
  Fill in code here
- ‘check if password is in the database
  Fill in code here
- ‘get username from database
  Fill in code here
- ‘display Hello username
  Fill in code here

- Remember that every if must have an End If after it. This concept is similar to HTML start and end tags.
- Capitalization matters! “Hello” is different from “hello”.

For More Information:

Unit 1 Section 4 Pre-Test

Please take this test before the lesson.

Name: ______________________________________

Directions: Please select a multiple choice answer that answers each question.

1. If `txtInput` is a textbox, and `lblValue` is a label, then what does this line of code do?

If `txtInput.Text = "Hello"` Then
   `lblValue.Text = "Hi. How are you?"
End If

   a. If the textbox says “hello”, then `lblValue` should display “Hi. How are you?”
   b. Change the text of the textbox to “Hi. How are you?”
   c. Change the text of the textbox to “Hello”
   d. If the textbox says “Hello”, then `lblValue` should display “Hi. How are you?”

2. What is wrong with the following lines of code?

If `lblbird.Text = "Life"` and `lblhero.Text = "Hero"` and `btnWorld.Text = "World"` Then
   `Lblenergy.Text = "121"
End If

   a. And is not allowed
   b. There is no End If
   c. `lblhero.Text = "Hero"` should be `lblhero.Text = "hero"
   d. `lblbird.Text = "Life"` should be `lblbird.Text = "bird"

3. If I type in “Hello” into `txtInput` (a textbox), then the text in the label `lblHello` does not change to “Hello”. Why?

If `txtInput.Text = "hello"` then
   `lblHello.Text = "Hello"
End If

   a. `hello` should be lowercase in order for the code to work
   b. the End If statement should be deleted
   c. `hello` in the first line should not be in double quotes
   d. `hello` in the first line should be in single quotes
Unit 1 Section 4 Post-Test

Please take this test after the lesson. Answers

Name: _________________________________________

Directions: Please select a multiple choice answer that answers each question.

4. If ‘txtInput is a textbox, and lblValue is a label, then what does this line of code do?

If txtInput.Text= “Hello” Then
lblValue.Text= “Hi. How are you?”
End If

a. If the textbox says “hello”, then lblValue should display “Hi. How are you?”
b. Change the text of the textbox to “Hi. How are you?”
c. Change the text of the textbox to “Hello”
d. If the textbox says “Hello”, then lblValue should display “Hi. How are you?”

5. What is wrong with the following lines of code?

If lblbird.Text=“Life” and lblhero.Text=“Hero” and btnWorld.Text=“World” Then
Lblenergy.Text=“121”

a. And is not allowed
b. There is no End If
c. lblhero.Text=“Hero” should be lblhero.Text=“hero”
d. lblbird.Text=“Life” should be lblbird.Text=“bird”

6. If I type in “Hello” into txtInput (a textbox), then the text in the label lblHello does not change to “Hello”. Why?

If txtInput.Text= “hello” then
lblHello.Text=“Hello”
End If

a. hello should be lowercase in order for the code to work
b. the End If statement should be deleted
c. hello in the first line should not be in double quotes
d. hello in the first line should be in single quotes
Unit 1 Section 5: Data Types
Author: Hadiya Harrigan
Date Created: 04-01-2014

Please take the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to teach you about different data types. You will be reintroduced to strings. You will be introduced to decimals, strings, and the Boolean data type. Also, you will learn how to convert between data types.

Overview:
Data Types: The word data type refers to how the data is stored.

- A **string** is a data type that is stored as text. For example, if you want the word Hello to appear on the screen, it is a string. Strings are referred to in double quotes (ex. “Hello”).
  - There are many object properties that require the use of strings.
    - Examples include:
      - Label1.Text
      - Button1.Text
      - Label1.ID
      - Button1.ID
      - Textbox1.Text
      - Textbox1.ID
    - When any string property is reassigned or compared in the script, then it must be set equal to a string set in double quotes:
      - For example, Label1.Text=“Hello” will write Hello in a label on the screen.
    - There are also many commands that use strings.
      - For example, Response.Write(“Hello”) will write out the word “Hello” on the line.
- An **integer** is a data type that stores integers. There are many categories of integers that tell the computer how much space to allot to store the integer. For example, Int32 and Int64 are both different types of integers.
  - Integers are commonly used for performing mathematical operations. You can add, subtract, multiply, divide, and do more to integers. However, you cannot do the same with strings. You cannot do mathematical operations using strings; strings must be converted to another data type such as a decimal or integer in order to perform mathematical operations.
  - Also, integers cannot be displayed as text. Integers must be converted to strings if the text must appear on a webpage.
- A **decimal** is a data type that stores decimals.
  - Decimals cannot be displayed as text. Decimals must be converted to strings if the text must appear on a webpage.
  - Decimals can be used to perform mathematical operations.
A **Boolean** data type is either True or False.

- Many object properties require the use of a Boolean data type.
  - Examples include:
    - Label1.Visible
    - Button1.Visible
    - Textbox1.Visible
  - When any string property is reassigned or compared in the script, then it must be set equal to either True or False. No double quotes are required.
    - Examples include:
      - Label1.Visible=True
      - Button1.Visible=False

*Note: Integers and Strings cannot be added together because they are the same data type*

**Conversion:** You often need to convert between data types. For example, you may need to convert a string into an int32 so that you can do mathematical operations with it.

The format of the statement to convert data types is:

```
Convert.ToDatatype(what you want to convert)
```

For example:

- Convert.ToInt32(Textbox1.Text)
  - This statement will convert the text in Textbox1 into a string.
- Convert.ToDecimal(Textbox1.Text)
  - This statement will convert the text in Textbox1 into a decimal.

**Problem Introduction:**

**Problem 1:** Create a multiplication calculator using the Int32 data type.

**Problem 2:** Create a multiplication calculator using the Decimal data type.

**Problem 3:** Create an addition calculator using the Int32 data type.

**Problem 4:** Create an addition calculator using the decimal data type.

**Problem Programming Steps:**

**Problem 1: Create a multiplication calculator using the Int32 data type.**

1. Create a new website. We recommend that you call the website “calculator”.
2. Create a new page. We recommend that you call the page “multiplicationint.aspx”. Remember not to use code behind.
3. Drag in the following objects in this format:

```
<body>
  <Label>Label</Label>
  <TextBox1>TextBox1</TextBox1>
  <Button>Button</Button>
  <label>Label</label>
</body>
```
4. Name the following objects as suggested. Therefore, if you are in ‘Source’ mode, change the ID and text properties as shown below:

```html
<asp:Label ID="lblTitle" runat="server" Text="Multiplication Calculator- Int32"></asp:Label>
<br />
<asp:TextBox ID="txt1" runat="server"></asp:TextBox>
<asp:Label ID="lblx" runat="server" Text="x"></asp:Label>
<asp:TextBox ID="txt2" runat="server"></asp:TextBox>
<asp:Button ID="btnMultiply" runat="server" Text="Multiply" />
<asp:Label ID="lblAnswer" runat="server" Text=""></asp:Label>
```

Therefore, in Design mode, the website will look like this:

```
<body>
<form id="form1" runat="Server">
</form>
</body>
```

5. Next, double click btnmultiply to create event when the object is clicked. In this sub, we must:

<table>
<thead>
<tr>
<th>Action/Step</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convert txt1.text to an int32</td>
<td>Convert.ToInt32(txt1.Text)</td>
</tr>
<tr>
<td>2. Convert txt2.text to an int32</td>
<td>Convert.ToInt32(txt2.Text)</td>
</tr>
</tbody>
</table>

Note that the * sign is used for multiplication and the / sign is used for division.
The following lines of code will perform these actions:

```csharp
<script runat="server">
    Protected Sub btnmultiply_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    End Sub
</script>

*Note that this website will only multiply integers not decimals.*

Problem 2: Create a multiplication calculator using the Decimal data type.

6. Create a new page called “multiplicationdecimal.aspx”. Make sure that you do not use code behind.
7. Repeat Steps 3-5. However, wherever the int32 data type is used, use the decimal datatype.

Individual Problems:
Try to solve these problems on your own. See the Tips section and Problem Programming Steps Section for help.

Problem 3: Create an addition calculator using the Int32 data type.

Problem 4: Create an addition calculator using the decimal data type.

Tips:
• Write out what you want to do using comments before you do it, or write out the code step by step as shown in the table in step 5.
• Make sure that you do math with decimals or integers.
• Make sure that you display strings.

For More Information:
Unit 1 Section 5 Pre-Test

Please take this test before the lesson.

Name: __________________________________

Please select the best data type to use for the following from the word bank.

Word Bank:
- Boolean
- Integer (Int32)
- String
- Decimal

12 _____
Hello _____
True _____

12.2 _____

Please choose a multiple choice answer:

1. Which line of code will convert the text in the text box txt1 into an integer?
   a. Convert.ToString(txt1)
   b. Convert.ToString(txt1.Text)
   c. Convert.ToInt32(txt1)
   d. Convert.ToInt32(txt1.Text)

2. Which line of code will convert the number 23 into text and display it in the text box hello?
   a. hello=Convert.ToInt32(hello)
   b. hello.text=Convert.ToString(23)
   c. hello.text=Convert.ToInt32(23)
   d. hello.text=Convert23totext

3. What is wrong with this statement? Button1 is a button.
   Button1.Text=Convert.ToInt32(“32”)
   a. Convert.ToInt32(“32”) converts “32” to an integer. However, the property Button1.Text requires the data type to be a string.
   b. 32 is in double quotes. It should be in single quotes.
   c. 32 is in double quotes. It should not have any quotes around it.
   d. Convert.ToInt(“32”) converts “32” to a string. However, the property Button1.Text requires the data type to be an integer.
Unit 1 Section 5 Post-Test
Please take this test after the lesson. Answers

Name: ___________________________________

Please select the best data type to use for the following from the word bank.

12 _____

Hello _____

True _____

12.2 _____

Word Bank:
- Boolean
- Integer (Int32)
- String
- Decimal

Please choose a multiple choice answer:

1. Which line of code will convert the text in the text box txt1 into an integer?
   a. Convert.ToString(txt1)
   b. Convert.ToString(txt1.Text)
   c. Convert.ToInt32(txt1)
   d. Convert.ToInt32(txt1.Text)

2. Which line of code will convert the number 23 into text and display it in the text box hello?
   a. hello=Convert.ToInt32(hello)
   b. hello.text=Convert.ToString(23)
   c. hello.text=Convert.ToInt32(23)
   d. hello.text=Convert23totext

3. What is wrong with this statement? Button1 is a button.
   Button1.Text=Convert.ToInt32(“32”)
   a. Convert.ToInt32(“32”) converts “32” to an integer. However, the property Button1.Text requires the data type to be a string.
   b. 32 is in double quotes. It should be in single quotes.
   c. 32 is in double quotes. It should not have any quotes around it.
   d. Convert.ToInt(“32”) converts “32” to a string. However, the property Button1.Text requires the data type to be an integer.
Unit 1 Section 6: Concatenation
Author: Hadiya
Date Created: 04-05-2014

Please take the Pre-Test before starting this lesson.

**Purpose:** The purpose of this lesson is to learn concatenation. By learning concatenation, you can make more complex websites.

**Overview:** Concatenation is an operation that joins strings together. To concatenate in ASP.net you can use a plus sign `+` or an and sign `&`.

See the table below for examples:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Code</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Add the strings “Hello” “ ” and “Goodbye”</td>
<td>“Hello” &amp; “ ” &amp; “Goodbye”</td>
<td>“Hello Goodbye”</td>
</tr>
<tr>
<td>2) Add the strings “Hello” and “Goodbye”</td>
<td>“Hello” &amp; “Goodbye”</td>
<td>“HelloGoodbye”</td>
</tr>
<tr>
<td>3) Add the strings “Hello” and “Goodbye”</td>
<td>“Hello” &amp; “ Goodbye”</td>
<td>“Hello Goodbye”</td>
</tr>
<tr>
<td>4) Add the strings “Hello ” and “Goodbye”</td>
<td>“Hello ” &amp; “Goodbye”</td>
<td>“Hello Goodbye”</td>
</tr>
<tr>
<td>5) Add the text in Textbox1 and Textbox2 Assume that the Textbox1.Text=”Hello” and Textbox2.Text=“Goodbye”</td>
<td>Textbox1.Text &amp; Textbox2.Text</td>
<td>“HelloGoodbye”</td>
</tr>
<tr>
<td>6) Add the text in Textbox1 a space and the text in Textbox2</td>
<td>Textbox1.Text &amp; “ ” &amp; Textbox2.Text</td>
<td>“Hello Goodbye”</td>
</tr>
<tr>
<td>7) Add the text in Textbox1, Textbox2, and a space</td>
<td>Textbox1.Text &amp; Textbox2.Text &amp; “ ”</td>
<td>“HelloGoodbye “</td>
</tr>
</tbody>
</table>

Notes: These notes refer to the numbered entries above.

1) Here “ ” is a space.
2) Because no space is included, there is no space between the added textboxes.
3) There is a space in the result because there is a space before Goodbye.
4) There is a space in the result because there is a space after Hello.
5) There is not space because “ ” was not added between the Textbox#.Text references.
6) Notice that when an object property is used, then there are no double quotes put around the property. See numbers 8 and 9 to see what happens when double quotes are put around an object property.
7) Order matters! Because the space was added at the end, there is no space between Hello and Goodbye, and the space is after hello and Goodbye.

8) When you put double quotes around an object’s property, the value of the property will not be replaced. Therefore, Textbox1.Text, the property, is different than “Textbox1.Text” the string.

9) This example is the same as number 8, except a space is added.

**Problem Introduction:**

**Problem 1:** Create a typing program with 6 keys: a, b, c, d, e, and space. Each key will be represented by a button. When the user clicks a key, the appropriate character will be added to a textbox. When the user is finished. They will click a done button. When the done button is clicked, the textbox will disappear, the 6 typing buttons will disappear, the done button will disappear, and the text in the textbox will be put into a label.

**Problem 2:** Present the user with two textboxes and a button. When the user clicks a button, add the strings in the two textboxes together. Put the final string into a label.

**Problem Programming Steps:**

**Problem 1:** Create a typing program with 6 keys: a, b, c, d, e, and space. Each key will be represented by a button. When the user clicks a key, the appropriate character will be added to a textbox. When the user is finished. They will click a done button. When the done button is clicked, the textbox will disappear, the 6 typing buttons will disappear, the done button will disappear, and the text in the textbox will be put into a label.

1. Create a new website. We suggest that you name the website “concatenationpractice”.
2. Create a new webpage. We suggest that you name the webpage “typingprogram.aspx”. Remember to create a page without code behind.
3. Create a page that has the following page layout. The ‘Source’ code and ‘Design’ view are provided for you. Make sure that you have the correct layout, IDs (names), and Text values. You may organize the page using <p></p> (paragraph tags), <br/> (line breaks), tables, and more. You may also use a combination. The following code is only one way to set up the user interface. There are many other ways to accomplish the same task.

**Design View:**

![Design View](image-url)
Source Code:

```html
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="txttitle" runat="server" Text="Typing Program"></asp:Label>
      <br />
      <br />
      <asp:TextBox ID="txttext" runat="server"></asp:TextBox>
      <br />
      <br />
      <asp:Label ID="lbltext" runat="server"></asp:Label>
      
      <asp:Button ID="btنا" runat="server" Text="ا" onclick="btنا_Click" />
      <asp:Button ID="btب" runat="server" Text="ب" onclick="btب_Click" />
      <asp:Button ID="btن" runat="server" Text="ن" onclick="btن_Click" />
      <br />
      <br />
      <asp:Label ID="btnc" runat="server" Text="c" onclick="btnc_Click" style="height: 26px" />
      <br />
      <br />
      <asp:Button ID="btند" runat="server" Text="d" onclick="btند_Click" />
      <asp:Button ID="btئ" runat="server" Text="ء" onclick="btئ_Click" />
      <br />
      <br />
      <asp:Button ID="btnspace" runat="server" Text="Space" onclick="btnspace_Click" />
      <br />
      <br />
      <br />
      <asp:Button ID="btnDone" runat="server" Text="Done" onclick="btnDone_Click" />
    </div>
  </form>
</body>
```

4. Double click each of the buttons while you are in ‘Design’ mode to create subroutines in ‘Source’ mode that occur when each of the buttons are clicked.

5. Type in the code for each button.
   a. For the letter buttons:
      - Each letter button must input a letter into the text box. However, it must not replace the entirety of the letters that were already in the text box.
      - If the string “c” was already in the text box, and the code for btnا was `txttext.Text= "ا"`, then the “c” would be entirely replaced, and only “أ” would remain.
      - In order to avoid this, we can add “أ” to the rest of the text in the textbox by using the code: `txttext.Text=txttext.Text & "أ"`. This line of code uses concatenation.
      - Using this knowledge, write the code for the letter buttons. The code is given in step 6 if you need help.
   b. For the space button:
      - A space is represented as “ ”. Use this fact and the information above to write the code for the space button.
      - The code is given in step 6 if you need help.
   c. For the done button:
• The problem says that these actions must happen when the done button is clicked:
  o The textbox will disappear
    ▪ This action is based on the property of visibility. Therefore, set the visible property to False.
  o The 6 typing buttons will disappear
    ▪ This action is based on the property of visibility. Therefore, set the visible property to False.
  o The done button will disappear
    ▪ This action is based on the property of visibility. Therefore, set the visible property to False.
  o The text in the textbox will be put into a label
    ▪ This action uses the text properties of each label. Therefore, set the text properties of each label equal to one another. When you do this, the order is: text property that should be reassigned= object text property that contains the desired text

6. Debug your application to see if it works. If it does not work, below is example working script code.

```csharp
<script runat="server">
Protected Sub btna_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & "a"
End Sub
Protected Sub btnb_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & "b"
End Sub
Protected Sub btnC_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & "c"
End Sub
Protected Sub btnd_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & "d"
End Sub
Protected Sub btne_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & "e"
End Sub
Protected Sub btnspace_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Text = txttext.Text & " "
End Sub
Protected Sub btnDone_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    txttext.Visible = False
    btna.Visible = False
    btnb.Visible = False
    btnC.Visible = False
    btnd.Visible = False
    btne.Visible = False
    btnspace.Visible = False
    btnDone.Visible = False
    lbltext.Text = txttext.Text
End Sub
</script>
```
Individual Problem:
Please see the Tips section for more help.

Problem 2: Present the user with two textboxes and a button. When the user clicks a button, add the strings in the two textboxes together. Put the final string into a label.

Tips:
- Name your objects appropriately, so that you understand what type of object and what the object does based on the name.
  - Ex. Txtfirstname.text might collect data on the person’s first name.
- Organize your user interface. You can use tables, line break tags, paragraph tags, spaces and more.

For More Information:
Unit 1 Section 6 Pre-Test
Please take this test before the lesson.

Name: ____________________________________

Directions: Please answer the following questions by using the given information to select a multiple-choice answer.

Given:
• Textbox1.Text="I have"
• Textbox2.Text= “many”
• Textbox3.Text= “questions”
• Textbox4.Text= “too”

1. What is the result of this concatenation? “I have” & Textbox1.Text & “too”
   a. “I have have too”
   b. “I have havetoo”
   c. “I have Textbox1.Text too”
   d. “I have I have too”

2. What is the result of this concatenation? “Textbox4.Text” & “too”
   c. “tootoo”
   d. “Textbox4.Texttoo”

3. What is the result of this concatenation? “Textbox4.Text.3.Text” & “many” & “” & “1”
   a. “too questions many 1”
   b. “too.3.Textmany 1”
   c. “Textbox4.Text.3.Textmany 1”
   d. “Textbox4.Text.3.Text many 1”

4. What is the result of this concatenation? Textbox1.Text & Textbox2.Text & Textbox3.Text
   a. “I have many questions”
   b. “123”
   c. “I have too many questions”
   d. “I have many questions”

5. What is the result of this concatenation?
   a. “I have many questions”
   b. “123”
   c. “I have too many questions”
   d. “I have many questions”
Unit 1 Section 6 Post-Test

Please take this test after the lesson. Answers

Name: _____________________________________________

Directions: Please answer the following questions by using the given information to select a multiple choice answer.

Given:

- Textbox1.Text="I have"
- Textbox2.Text= "many"
- Textbox3.Text= “questions”
- Textbox4.Text= “too"

1. What is the result of this concatenation? “I have” & Textbox1.Text & “too"
   a. “I havelhavetoo”
   b. “I havel havetoo”
   c. “I have Textbox1.Text too”
   d. “I have I have too”

2. What is the result of this concatenation? “Textbox4.Text” & “too”
   c. “tootoo”
   d. “Textbox4.Texttoo”

3. What is the result of this concatenation? “Textbox4.Text.3.Text” & “many” & “” & “1”
   a. “too questions many 1”
   b. “too.3.Textmany 1”
   c. “Textbox4.Text.3.Textmany 1”
   d. “Textbox4.Text.3.Text many 1”

4. What is the result of this concatenation? Textbox1.Text & Textbox2.Text & Textbox3.Text
   a. “I have many questions”
   b. “123”
   c. “I have too many questions”
   d. “I havemanyquestions”

5. What is the result of this concatenation?
   a. “I have many questions”
   b. “123”
   c. “I have too many questions”
   d. “I havemanyquestions”
Unit 1 Section 7: Multi-Page Websites
Author: Hadiya
Date Created: 4-12-2014

Please take the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to teach you how to link multiple pages into a website. Also, color changes will be introduced.

Overview:

Linking Webpages:
Below, we will describe three ways of linking webpages together.

1. You can link webpages using html tags.
   The format of the html tag is: `<a href="url">Link text</a>`
   For example, you can link to google by using the tag:
   `<a href="http://www.google.com"> Google </a>`
   This is called an anchor tag.
   *Make sure that you use the full URL and put the URL in double quotes

2. You can create a button, and in its associated subroutine, use the Response.Redirect("url") command.
   For example: Response.Redirect("http://www.google.com")
   *Make sure that you use the full URL and put the URL in double quotes

3. You can create a menu.
   Steps to create a menu:
   1. Drag and drop a menu from the navigation section of the toolbox.
   2. Click the arrow that appears. Next, click Edit Menu Items.
3. Add new menu items by clicking the button below:

4. Change the following properties:
   a. Text and Value: If you change one, then the other may automatically change. In many cases, these two properties are the same.
   b. Navigate Url: If you want to link to Google, or another outside webpage, type the whole URL without double quotes.
However, if you want to link to another page within the application, click the … button. Then select a page that you want from the provided list of pages.

5. Keep adding new items until your menu is complete. Click OK when you are finished.

Auto formatting menu colors:
1. Click the arrow on the side of the menu.
2. Select Auto format
3. Select a template from the list. Then click OK.
Background Color:
Just like you can change the properties of a textbox or button, you can also change the properties of the webpage itself. One property that you can change is bgcolor or background color.

1. Go to the properties pane and select the DOCUMENT.

2. Go to the bg color property and click the ... button.
3. Choose a color. Notice that the value is a number in **Hex (Hexadecimal)** that represents the color that you selected. There are some links to learn more about Hex in the for more information section.

![Color Selection](image)

4. Click OK.

**Problem Introduction:**

**Problem 1:** Create a webpage with 2 buttons and 2 links that connect to different websites. Use your favorite color as the background color.

**Problem 2:** Create a webpage with a menu that connects one of your previous applications together. Change the background color for each of the webpages.

**Problem 3:** Create a new website. Create 5 different pages with different background colors. Make a menu on each page that links to all 5 pages.

**Problem Programming Steps:**

**Problem 1: Create a webpage with 2 buttons and 2 links that connect to different websites. Use your favorite color as the background color.**

1. Create a new website. We recommend that you call the website ‘navigation’.
2. Create a new webpage. We recommend that you call the webpage ‘links’.
3. Create a new 2x2 table. There are directions for creating a table in Unit 1 Lesson 4.
4. In the left column, put one button in each row. Therefore, there are two buttons in the left column. Name the buttons appropriately. Change the text to the website that you plan to link the buttons to.
5. In the right column, type link in each cell. This will serve as a marker so that we know where the link goes in the ‘Source’ code.
6. Double click each button to create a subroutine. Use the information from the overview to create a button that redirects you to your selected websites. In this example, we use Google and Youtube.

```csharp
<%<script runat="server">

Protected Sub btnGoogle_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Response.Redirect("http://www.google.com")
End Sub

Protected Sub btnYoutube_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Response.Redirect("http://www.youtube.com")
End Sub
</script>%
```

7. Go to the ‘Source’ code. Find the first instance where you typed Link. Replace this with the html in the overview that can be used to create a link. In our example, we use Google and Youtube.
8. Look at the ‘Design’ mode to see how the links appear.
9. Debug your website to see if all of the links work.
10. Change the background color. (See the overview).
11. Debug your website again.

Problem 2: Create a webpage with a menu that connects one of your previous applications together. Change the background color for each of the webpages.

12. Choose a website that you have previously created that has at least 3 pages. We suggest that you use the Unit 1 Section 5 Data Types lesson. We suggested that you named this website ‘calculator’.
13. Create a menu.
14. Use the steps from the overview to create an item for each page in the problem.
15. Copy and paste this menu onto each of the pages.
16. Debug and navigate around each of the pages.
17. Change the background colors of each of the pages.
18. Debug.

Individual Problem:
Make sure to read the Tips section before starting on this problem.

Problem 3: Create a new website. Create 5 different pages with different background colors. Make a menu on each page that links to all 5 pages.
Tips:
- To put a menu on multiple pages, you only need to make the menu once. You can copy and paste the menu onto each of the other pages.

For More Information:
Unit 1 Section 7 Pre-Test

Please take this test before the lesson.

Name: _______________________________________

Answer these questions by selecting a multiple choice answer choice.

1. When a menu is created, the property for the url of an item is:
   a. Text
   b. WebsiteName
   c. Url
   d. NavigateUrl

2. When a menu is created, the property for the displayed text of an item is:
   a. Text
   b. WebsiteName
   c. Url
   d. NavigateUrl

3. In the following HTML tag, the a stands for what?
   \(<a\ href="http://www.google.com"> Google </a>\)
   a. asset
   b. anchor
   c. alternate
   d. asymptote

4. What is the property that is used to set the background color of the document?
   a. Background
   b. Color
   c. Bgcolor
   d. Back

5. Which data type is used for colors?
   a. Octal
   b. String
   c. Int32
   d. Hexidecimal

6. What can you use to quickly set a color theme for a menu?
   a. autoformat
   b. autocolor
   c. formatcolor
   d. formatbackground
Unit 1 Section 7 Post-Test

Please take this test after the lesson. Answers

Name:_________________________________

Answer these questions by selecting a multiple choice answer choice.

1. When a menu is created, the property for the url of an item is:
   a. Text
   b. WebsiteName
   c. Url
   d. NavigateUrl

2. When a menu is created, the property for the displayed text of an item is:
   a. Text
   b. WebsiteName
   c. Url
   d. NavigateUrl

3. In the following HTML tag, the a stands for what?
   <a href="http://www.google.com"> Google </a>
   a. asset
   b. anchor
   c. alternate
   d. asymptote

4. What is the property that is used to set the background color of the document?
   a. Background
   b. Color
   c. Bgcolor
   d. Back

5. Which data type is used for colors?
   a. Octal
   b. String
   c. Int32
   d. Hexidecimal

6. What can you use to quickly set a color theme for a menu?
   a. autoformat
   b. autocolor
   c. formatcolor
   d. formatbackground
Unit 1 Section 8: Variables
Author: Hadiya
Date Created: 4-12-2014

Please take the Pre-Test before starting this lesson.

Purpose: The purpose of this lesson is to teach you about variables.

Overview:

A. What is a variable? A variable is a word that can stand in for something and assumes its value. A variable can have any data type: Int32, Int16, String, Boolean, Decimal etc..

B. To create a variable use the following pattern:
Dim nameofvariable as datatype
For example: Dim myinteger as string
*Dim means dimension

C. To refer to a variable:
This variable will now be referred to as myinteger
For example: myinteger=2 will set the variable equal to 2
*You may get an error if you do not set the variable equal to something.
*The statement myinteger=“2” is also incorrect because you are setting an integer equal to a string.

D. Examples:
Because the variable can stand in for numbers, you can do math with integer variables and you can display string variables, just as if you had typed the actual text.
Assume:

- myinteger is a variable of the datatype Int32 that has the value 2.
- anotherinteger is a variable of the datatype Int32 that has the value 5.
- mystring is a variable of the datatype String that has the value “PDF”
- anotherstr is a variable of the datatype String that has the value “ACT”

Concatenation:

<table>
<thead>
<tr>
<th>#</th>
<th>Code</th>
<th>Value (after concatenation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mystring &amp; anotherstr</td>
<td>“PDFACT”</td>
</tr>
<tr>
<td>2</td>
<td>mystring &amp; &quot; &quot; &amp; anotherstr</td>
<td>“PDF ACT”</td>
</tr>
<tr>
<td>3</td>
<td>“mystring” &amp; “anotherstr”</td>
<td>“mystringanotherstr”</td>
</tr>
<tr>
<td>4</td>
<td>“mystring” &amp; “ “ &amp; “anotherstr”</td>
<td>“mystring anotherstr”</td>
</tr>
</tbody>
</table>

Note: For examples 3 and 4, note that the variable cannot be referred to with double quotes around them. If double quotes are used, then the name of the variable will be interpreted as a text.
Table of Contents

Math:

<table>
<thead>
<tr>
<th>#</th>
<th>Code</th>
<th>Value (after math is done)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>myinteger+2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>“myinteger” + 2</td>
<td>error</td>
</tr>
<tr>
<td>3</td>
<td>myinteger + anotherint</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>myinteger * anotherint</td>
<td>10</td>
</tr>
</tbody>
</table>

Note: For example 2, a string cannot be added to a number. Therefore, this results in an error.

Problem Introduction:

Problem 1: Create a division calculator using int32 variables.

Problem 2: Add two strings together using variables. Put a space in the middle of the two strings.

Problem 3: Create a subtraction calculator using decimal variables.

Problem Programming Steps:

Problem 1: Create a division calculator using int32 variables.

1. Create a new website. We recommend that you name the website ‘variables’.
2. Create a new webpage. Remember to disable code behind. We recommend that you name the webpage ‘divisioncalc.aspx’.
3. Layout these objects to set up a calculator. The ‘Design’ and ‘Source’ code are located below.

```html
<body>
  <form method="get">
    <div>
      <input type="number" id="num1" placeholder="Enter first number" required>
      <input type="number" id="num2" placeholder="Enter second number" required>
      <button type="submit" id="divide">Divide</button>
    </div>
    <p id="result"></p>
  </form>
  <script>
    const num1 = document.getElementById('num1');
    const num2 = document.getElementById('num2');
    const result = document.getElementById('result');
    num1.addEventListener('input', () => {
      let num1Val = num1.value;
      num2.addEventListener('input', () => {
        let num2Val = num2.value;
        if (num1Val === '' || num2Val === '') {
          result.textContent = 'Please enter numbers.';
        } else {
          result.textContent = num1Val / num2Val;
        }
      });
    });
  </script>
</body>
```
4. Double click the button to create a subroutine that runs when the button is clicked.
5. In this subroutine, insert code that does the following actions:
   a. Create 3 int32 variables:
      i. Make a variable for the first input
      ii. Make a variable for the second input
      iii. Make a variable for the answer
   b. Assign the first input and second input values from the textboxes. Remember that the textbox text is a string. Therefore, the string must be converted to an integer.
   c. Divide the numbers, the value should be set as the answer variable.
   d. Print out the answer variable in a label. Because this is an integer, you must convert the integer to a string before setting it equal to the text of the label.

If you need help, the code for the button is below:

```vbnet
Protected Sub btndivide_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    'create variables
    Dim number1 As Int32
    Dim number2 As Int32
    Dim answer As Int32

    'assign the value of the variables
    number1 = Convert.ToInt32(txt1.Text)
    number2 = Convert.ToInt32(txt2.Text)

    'divide the numbers
    answer = number1 / number2

    'print out the answer
    lblanswer.Text = Convert.ToString(answer)
End Sub
```
6. Debug to make sure that your website works.
Problem 2: Add two strings together using variables. Put a space in the middle of the two strings.

7. Create a new webpage. Remember to disable code behind. We recommend that you name the webpage 'stringvariables.aspx'.

8. Layout these objects. The ‘Design’ and ‘Source’ code are located below.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="lbltitle" runat="server" Text="Concatenation with Variables"></asp:Label>
      <asp:TextBox ID="txt1" runat="server"></asp:TextBox>
      <asp:TextBox ID="txt2" runat="server"></asp:TextBox>
      <asp:Button ID="btnConcatenate" runat="server" Text="Concatenate" onclick="btnConcatenate_Click" />
      <asp:Label ID="lblanswer" runat="server" Text=""></asp:Label>
    </div>
  </form>
</body>
</html>
```

9. Double click the button to create a subroutine that runs when the button is clicked. Notice that when you double click the button, a new property in the code called OnClick appears. The value of OnClick is “btnConcatenate_Click” which is the same name of the subroutine in the script. Therefore, when you click the button, the button looks at its OnClick property and is referred to the subroutine that is the value of the OnClick property.

```
<asp:Button ID="btnConcatenate" runat="server" Text="Concatenate" onclick="btnConcatenate_Click" />
```

10. In the subroutine, program the following actions:
   a. Create 3 variables. The data type should be a string.
      i. A variable for the first input
      ii. A variable for the second input
      iii. A variable for the output/answer
   b. Set the variable for the inputs equal to their respective textbox text.
c. Add the strings together. Remember to put a space in the middle. Put the result in the answer variable.
d. Put the answer variable into a label.

If you need help, first look at the overview section. Also, the code for the button is below. However, please attempt to write the code and debug the errors before referring to the code.

```vbnet
Protected Sub btnConcatenate_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    'create variables
    Dim input1 As String
    Dim input2 As String
    Dim output As String

    'get values of the strings
    input1 = txt1.Text
    input2 = txt2.Text

    'add the strings
    output = input1 & " " & input2

    'output into the label
    lblanswer.Text = output
End Sub
```

11. Debug the webpage.

**Individual Problem:**

The following problem is an individual problem. Read the Tips section before attempting this problem.

**Problem 3: Create a subtraction calculator using decimal variables.**

**Tips:**
- Go back and read the problem when you are done to make sure that you have finished the entire problem.

**For More Information:**
Unit 1 Section 8 Pre-Test

Please take this test before the lesson.

Name: __________________________________

Directions: Determine what the values of the following lines of code are by using the given information. Select a multiple choice answer.

Given information:
- Variables
  - str1 is a string. The value of the string is “My name is”
  - str2 is a string. The value of the string is “Ben”
  - str3 is a string. The value of the string is “What is your name?”
  - str4 is a string. The value of the string is “Allie”
  - int1 is an Int32. The value is 32.
  - int2 is an Int32. The value is 8.
- Textboxes
  - Textbox1.Text = “1”
  - Textbox2.Text = “Mouse”

1. “str1” & “ “ & “str2”
   a. “Mynames Ben”
   b. “My name is Ben”
   c. “str1str2”
   d. “str1 str2”
2. str1 & “ “ & str2
   a. “Mynames Ben”
   b. “My name is Ben”
   c. “str1str2”
   d. “str1 str2”
3. str3 & “ “ & str1 & str4
   a. “What is your name?”
   b. “What is your name? Mynamesstr4”
   c. “What is your name? My name isAllie”
   d. “What is your name? My name isAllie”
4. Convert.ToInt32(Textbox1.Text) + int1 + int2
   a. “41” (the string)
   b. 41 (the integer)
   c. “fourty one”
   d. “Textbox1.Text328”
5. “Convert.ToInt32(Textbox1.Text) + int1 + int2”
   a. “Convert.ToString(Textbox1.Text) + int1 + int2”
   b. “41” (the string)
   c. 41 (the integer)
   d. “Textbox1.TextMouse”
Unit 1 Section 8 Post-Test
Please take this test after the lesson. **Answers**

Name: ______________________________________

Directions: Determine what the values of the following lines of code are by using the given information. Select a multiple choice answer.

Given information:

- **Variables**
  - `str1` is a string. The value of the string is “My name is”
  - `str2` is a string. The value is “Ben”
  - `str3` is a string. The value is “What is your name?”
  - `str4` is a string. The value is “Allie”
  - `int1` is an Int32. The value is 32.
  - `int2` is an Int32. The value is 8.

- **Textboxes**
  - `Textbox1.Text` = “1”
  - `Textbox2.Text` = “Mouse”

1. “str1” & “ “ & “str2”
   - e. “Mynameis Ben”
   - f. “My name is Ben”
   - g. “str1str2”
   - h. “str1 str2”

2. `str1` & “ “ & `str2`
   - a. “Mynameis Ben”
   - b. “My name is Ben”
   - c. “str1str2”
   - d. “str1 str2”

3. `str3` & “ “ & `str1` & `str4`
   - a. “What is your name?”
   - b. “What is your name?Mynameisstr4”
   - c. “What is your name?My name isAllie”
   - d. “What is your name? My name isAllie”

4. `Convert.ToInt32(Textbox1.Text) + int1 + int2`
   - a. “41” (the string)
   - b. 41 (the integer)
   - c. “fourty one”
   - d. “Textbox1.Text328”

5. “`Convert.ToInt32(Textbox1.Text) + int1 + int2“
   - a. “`Convert.ToString(Textbox1.Text) + int1 + int2”
   - b. “41” (the string)
   - c. 41 (the integer)
   - d. “Textbox1.TextMouse”
Unit 1 Section 9: Radio Button Lists, Check Box Lists, and DropDownLists

Author: Hadiya
Date Created: 4-12-2014

Please take the Pre-Test before starting this lesson.

Purpose: The purpose of this lesson is to introduce you to three more objects: Radio Button Lists, Check Box Lists, and Dropdown Lists.

Overview:
Radio Button Lists:
A radio button list is an object that allows you to select one of many given choices. Next to each choice is a circle, and next to the selected option there is a green dot in the circle.

To create a radio button list, drag in the object from the toolbox. Next, click the arrow on the right and Edit Items. The resulting window will enable you to edit the items in the list. This is very similar to the menu in the multipage website lesson (Unit 1 Section 8). However, no navigate url property is needed. Simply reset the text and value properties.

Therefore, when referring to the selected item of a radio button list, use the properties ‘radiobuttonlistname.text’ or ‘radiobuttonlistname.selectedvalue’.

Checkbox List
A checkbox list is similar to a radio button list. However, the shape next to each choice is a square, many items can be selected, and a green check appears next to each selected item.

Use the same steps as the radio button list to create a checkbox list.

To refer to a checkbox list, if you use the properties ‘checkboxlistname.text’ or ‘checkboxlistname.selectedvalue’ you will only get text or value of the first selected checkbox. Therefore, you can use ‘checkboxlistname.items(#).Text’ to get the text. Also to check if a certain checkboxlist item is checked use ‘checkboxlistname.items(#).selected’ and set it equal to a Boolean
value (see Unit 1 Lesson 5 Data Types for more about Boolean values; also see example below) In these examples, the number sign stands in place of the index of the item. The index numbers were given to you when you edited the items of the textbox. The numbers start at zero and increase.

Example:

If radiobuttonlistname.items(0).selected=true then
End If

Dropdown List
A drop down list is similar to a radio button list. A drop down list is an object that allows you to select one of many given choices. However, only one choice is displayed at a time. You need to click the arrow so that the rest of the choices drop down.

Use the same steps as the radio button list to create a drop down list.

To refer to a drop down list, use the properties ‘radiobuttonlistname.text’ or ‘radiobuttonlistname.selectedvalue’.

Other
- If you do not want to type for example Label1.Text=Label1.Text & “anotherstring”, you can instead use this line of code: Label1.Text += “anotherstring”. The result of this line of code is to concatenate “anotherstring” onto the end of what was already in the label.
- You can include break tags within segments of code that include double quotes. For example:
  o Label1.Text = “hello <br/> goodbye”. This will put hello and goodbye on separate lines.
These lines of code will produce two lines of text at the top of the page due to the break tags.

Problem Introduction:

Problem 1:

Create a website that allows the user to fill out a high school form. Organize the form in a table, or using <br />, <p></p>, etc.
- Use a text box for the user to write their first name
- Use a text box for the user to write their middle name
- Use a text box for the user to write their last name
- Use a radio button list for the user to select a grade (9-12)
- Use a drop down list for the user to select t-shirt size (S,M,L,XL,XXL)
- Use a drop down list for the user to select an advisor (Mr. Hill, Mrs. Prince, Mr. Murph, Mr. Harrigan, Mrs. Asante)
- Use a check box list for the user to select their classes (Math, Science, Spanish, History, English)
After a submit button is clicked, display the following statement (fill in the blanks with the information that you collected):

- Your name is (first name) (middle name) (last name).
- You are in Grade (grade).
- Your t-shirt size is (size).
- Your advisor is (advisor name).
- You want to take the following classes:
  (classes)

Problem 2:

Create a madlib. See (http://www.itsamadlibsworld.com/) for examples.

Allow users to use dropdownlists, radiobuttonlists, and checkboxlists to fill in random words. Reveal the story in a label after the submit button is clicked.

Problem Programming Steps:

Problem 1:

1. Create a new website. We recommend that you call this website ‘otherobjects’.
2. Create a new page without code behind. We recommend that you name this page ‘highschoolform’
3. Create a page with the following objects. Double click all buttons to create a subroutine that runs when the button is clicked. The ‘Source’ code and ‘Design’ mode views are given below.
In this version of the source code, the objects that take up more space are minimized. The code from these sections will be included later.

```html
<body>
  <form id="form1" runat="server">
    <div>
      <h4>High School Form:</h4>
      First Name:<br />
      <asp:TextBox ID="txtfname" runat="server"></asp:TextBox><br />
      Middle Name:<br />
      <asp:TextBox ID="txtmname" runat="server"></asp:TextBox><br />
      Last Name:<br />
      <asp:TextBox ID="txtlname" runat="server"></asp:TextBox><br />
      Grade:<br />
      <asp:RadioButtonList ID="rbgrade" runat="server"></asp:RadioButtonList>
      T-Shirt size:<br />
      <asp:DropDownList ID="ddlshirt" runat="server"></asp:DropDownList>
      Advisor:<br />
      <asp:DropDownList ID="ddladvisor" runat="server"></asp:DropDownList>
      Select the subjects that you would like to take:<br />
      <asp:CheckBoxList ID="cblsubject" runat="server" onclick="cblSubmit_Click" Text="Submit" />
    </div>
  </form>
  <asp:Label ID="lbloutput" runat="server"></asp:Label>
</div>

Below are the missing lines of code:

```html
<asp:RadioButtonList ID="rbgrade" runat="server">
  <asp:ListItem>9</asp:ListItem>
  <asp:ListItem>10</asp:ListItem>
  <asp:ListItem>11</asp:ListItem>
  <asp:ListItem>12</asp:ListItem>
</asp:RadioButtonList>
<asp:DropDownList ID="ddladvisor" runat="server">
  <asp:ListItem>Mr. Hill</asp:ListItem>
  <asp:ListItem>Mrs. Prince</asp:ListItem>
  <asp:ListItem>Mr. Murph</asp:ListItem>
  <asp:ListItem>Mr. Harrison</asp:ListItem>
  <asp:ListItem>Mrs. Asante</asp:ListItem>
</asp:DropDownList>
<asp:CheckBoxList ID="cblsubject" runat="server">
  <asp:ListItem>Math</asp:ListItem>
  <asp:ListItem>Science</asp:ListItem>
  <asp:ListItem>History</asp:ListItem>
  <asp:ListItem>English</asp:ListItem>
</asp:CheckBoxList>

4. Program the button to do the following when it is clicked:
   a. Print out “Your name is ” (first name) (middle name) (last name) into the label
   b. Line break
   c. Print out “You are in Grade ” (grade) into the label
   d. Line break
   e. Print out “Your t-shirt size is ” (size) into the label
   f. Line break
   g. “Your advisor is ” (advisor name) into the label
   h. Line break

See the code below and the overview if you need help.
5. Because multiple items in the checkbox can be selected, the code for the classes in the checkbox cannot be done this way. Instead, use these steps:
   a. Print out “You want to take the following classes:” into the label
   b. Line break
   c. Create a series of 5 if statements that check whether or not each list item is checked. If the list item is checked, then print out the subject and go to the next line.

See the code below and the overview if you need help:

```vbnet
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    lbloutput.Text = "Your name is: " & txtfname.Text & " " & txtlname.Text & ".<br/>
    lbloutput.Text += "You are in grade: " & rblgrade.Text & "<br/>
    lbloutput.Text += "Your t-shirt size is: " & dtltshirt.Text & "<br/>
    lbloutput.Text += "Your advisor is: " & dddladvisor.Text & "<br/>

    'subjects
    lbloutput.Text += "You want to take the following classes: <br/>

    If cblsubject.Items(0).Selected = True Then
        lbloutput.Text += "Math<br/>
    End If
    If cblsubject.Items(1).Selected = True Then
        lbloutput.Text += "Science<br/>
    End If
    If cblsubject.Items(2).Selected = True Then
        lbloutput.Text += "Spanish<br/>
    End If
    If cblsubject.Items(3).Selected = True Then
        lbloutput.Text += "History<br/>
    End If
    If cblsubject.Items(4).Selected = True Then
        lbloutput.Text += "English<br/>
    End If

End Sub
```

6. Debug your application to test it.
Individual Problem:
Do this problem on your own. Read the Tips section below before starting.

Problem 2:

Create a madlib. See (http://www.itsamadlibsworld.com/) for examples.

Allow users to use dropdownlists, radiobuttonlists, and checkboxlists to fill in random words. Reveal
the story in a label after the submit button is clicked.

Tips:
● We recommend that you use the following naming convention.
  ○ Start every name (ID property) with the code for the object type
    ▪ Button (btn)
    ▪ Textbox (txt)
    ▪ Label (lbl)
    ▪ Radiobuttonlist (rbl)
    ▪ Checkboxlist (cbl)
    ▪ Dropdown List (ddl)
  ○ End with a name that describes the purpose of the object
  ○ Examples:
    ▪ BtnSubmit
    ▪ lbl FName
    ▪ txt LName
    ▪ BtnGo
    ▪ txtinput

For More Information:
Unit 1 Section 9 Pre-Test

Please take this test before the lesson.

Name: ______________________________________

Explain what the following lines of code do by selecting a multiple choice answer and referring to the given information.

Given information:

- cblactivities is a checkboxlist
  - cblactivities.items(0).selected=true
  - cblactivities.items(1).selected=false
  - cblactivities.items(2).selected=true
  - cblactivities.items(0).text="Golf"
  - cblactivities.items(1).text="Chess"
  - cblactivities.items(2).text="Choir"
- ddlname is a dropdownlist
  - ddlname.selectedvalue="Brianna Martin"
- rblsection is a radiobuttonlist
  - rblsection.selectedvalue="5"
- lblvalue is a label

1. Response.Write( Convert.ToString(25 + Convert.ToInt32(rblsection.selectedvalue)))
   a. Write “25 + convert.toint32(rblsection.selectedvalue)” at the top of the page.
   b. Write “30” at the top of the page
   c. Write “30” at the bottom of the page
   d. Write “25 at the top of the page

2. lblvalue.text= “Hello”
   lblvalue.text+= cblactivities.items(0).Text & “<br/>”
   lblvalue.text= “Goodbye”
   What does lblvalue.text equal after this code runs?
   a. “Goodbye”
   b. “HelloGolf”
      Goodbye”
   c. “HelloGolfGoodbye”
   d. “HelloGoodbye”

3. lblvalue.text = cblactivities.items(2).text
   lblvalue.text += ddlname.selectedvalue
   What does lblvalue.text equal after this code runs?
   a. “Choir”
   b. “Brianna Martin”
   c. “Choir Brianna Martin”
   d. “ChoirBrianna Martin”
Unit 1 Section 9 Post-Test

Please take this test after the lesson. Answers

Name:__________________________________

Explain what the following lines of code do by selecting a multiple choice answer and referring to the given information.

**Given information:**
- cblactivities is a checkboxlist
  - cblactivities.items(0).selected=true
  - cblactivities.items(1).selected=false
  - cblactivities.items(2).selected=true
  - cblactivities.items(0).text="Golf"
  - cblactivities.items(1).text="Chess"
  - cblactivities.items(2).text="Choir"
- ddnname is a dropdownlist
  - ddnname.selectedvalue="Brianna Martin"
- rblsection is a radiobuttonlist
  - rblsection.selectedvalue="5"
- lblvalue is a label

1. Response.Write( Convert.ToString(25 + Convert.ToInt32(rblsection.selectedvalue)))
   a. Write “25 + convert.toint32(rblsection.selectedvalue)” at the top of the page.
   b. Write “30” at the top of the page
   c. Write “30” at the bottom of the page
   d. Write “25 at the top of the page

2. lblvalue.text= “Hello”
   lblvalue.text+= cblactivities.items(0).Text & “<br/>”
   lblvalue.text= “Goodbye”
   What does lblvalue.text equal after this code runs?
   e. “Goodbye”
   f. “HelloGolf”
   G. “HelloGolfGoodbye”
   H. “HelloGoodbye”

3. lblvalue.text = cblactivities.items(2).text
   lblvalue.text += ddnname.selectedvalue
   What does lblvalue.text equal after this code runs?
   a. “Choir”
   b. “Brianna Martin”
   c. “Choir Brianna Martin”
   d. “ChoirBrianna Martin”
Unit 1 Section 10: Loops
Author: Hadiya
Date Created: 04-12-2014

Please take the Pre-Test before beginning the lesson.

Purpose: The purpose of this lesson is to teach you about loops. Loops run a segment of code over and over again until they are told to stop. Also, information about counters and operators is included.

Overview:
We will introduce you to 3 types of loops: do...loop while; do...loop until; for...next.

Counter:
Many loops take advantage of the concept of a counter. A counter is a variable (normally an Int32) that increases or decreases by a certain amount. For example, if you want to run something 10 times, create a counter (Int32 variable). You can name it whatever you want, however, we have chosen to name this variable counter.

Dim counter as int32

Set the counter equal to ten.

counter = 10

By using a loop that subtracts 1 every time the code runs, we can run a set of code 10 times.

Operators
An operator is a symbol used for comparison of two entities. For example, in an if statement, you may ask:

If TextBox1.Text=TextBox2.Text Then
Response.Redirect("http://www.google.com")
End If

In this example, the equal sign is the operator.

Another operator is ‘<>’. This means not equal to. Another way to write not equal to is ‘!=’.


The equal sign is another operator, just like <> and !=. Other operators include ‘<=’ (less than or equal to), ‘>=’ (greater than or equal to), ‘<’ (less than), and ‘>’ (greater than).

See http://msdn.microsoft.com/en-us/library/cey92b0t.aspx for a list of more operators.
**Do...Loop While**

This loop will do something while a certain condition is true. The format of this is:

```
Loop
[Insert Code that will run in the loop]
Loop While [Insert a condition]
```

For example, I may want to print out ‘Hello World’ multiple times while the counter is not equal to ten. The code for this is:

```vbs
Dim counter as Int32
Counter=10
Loop
Response.Write(“Hello World”)
counter=counter-1
Loop While counter <> 10
```

*Do not forget to include ‘counter=counter-1’. Without this line of code, you will create a loop that runs forever.*


**Do...Loop Until**

Another way to do the same operation is using a do... loop until loop. This loop will do something until a condition is met. The general format of this loop is:

```
Loop
[Insert Code that will run in the loop]
Loop Until [Insert a condition]
```

For example, you can print out ‘Hello World’ multiple times until the counter equals 1. The code for this is:

```vbs
Dim counter as Int32
Counter=10
Loop
Response.Write(“Hello World”)
counter=counter-1
Loop Until counter = 10
```

For...Next
Lastly, the For...Next Loop allows you to do something in between two values for a counter. The general format is:

```
For counter As datatype = start# to end#
[Insert Code that will run in the loop]
Next counter
```

Note that this loop does not need a statement in the code that will increment the counter variable. The loop will increment the countervariable by 1 each time the code runs.

For example, you can print out ‘Hello World’ multiple times while the counter is in between 1 and 10. The code for this is:

```
For counter As Int32= 1 to 10
Response.Write(“Hello World”) 
Next counter
```


Random Numbers
To create a random number generator, use this formatting:

```
variable = Int(Rnd() * #)
```

This code will generate numbers from 0 to #-1. For example, you can generate a random number that is between 0 and 4 by using the code:

```
Dim randomnumber as Int32
randomnumber=Int(Rnd() * 5)
```

However, if you want to generate a number from 1 to 5, use this code:

```
Dim randomnumber as Int32
randomnumber=Int(Rnd() *5)+1
```

The lowest number will now be 1 (because you added one to the previous lowest number, 0). The highest number will not be 6 (because you added one to the previous highest number, 4).

**Problem Introduction:**

**Problem 1:** Create a website that rolls a die until it reaches the number six. Print out each number that is rolled. Use a random number generator and a Do... Loop While loop.

**Problem 2:** Create a website that rolls a die until it reaches the number six. Print out each number that is rolled. Use a random number generator and a Do... Loop Until loop.
Problem 3: Create a website that counts from 1 to 10. Use a For...Next Loop.

Problem Solving Steps:

Problem 1: Create a website that rolls a die until it reaches the number six. Print out each number that is rolled. Use a random number generator and a Do... Loop While loop.

1. Create a new website. We recommend that you call this website diceandloops.
2. Create a new webpage. We recommend that you call this webpage doloopwhiledice.aspx.
3. This webpage can be created without the use of objects by using Response.Write(). Therefore, there is no need to drag in any objects from the toolbox.
4. Double click the document to create a sub page load subroutine that will run as soon as the page runs. Therefore, all important code will be in the script section of the page.
5. In the sub, write the code for the dice. Remember to increment your counter inside the loop. Remember to generate a random number for the dice. The number should be between 1 and 6. Try to code this by yourself before looking at the example code below:

   ```vb
   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
   Dim diceroll As Int32
   Response.Write("Test begin. We'll keep trying until we get a six. " & "<br/>")
   Do
   diceroll = Int(Rnd() * 6) + 1
   Response.Write("You rolled a " & Convert.ToString(diceroll) & "<br/>")
   Loop While diceroll <> 6
   Response.Write("Got it. Press page refresh to try again."")
   End Sub
   </script>
   
   6. Debug your application.

Individual Problems:
Do the following problems on your own. Please read the Tips section before starting. It may also be helpful to reread the overview section and look at the code for problem 1 if you get stuck.

Problem 2: Create a website that rolls a die until it reaches the number six. Print out each number that is rolled. Use a random number generator and a Do... Loop Until loop.

Problem 3: Create a website that counts from 1 to 10. Use a For...Next Loop.

Tips:
- For problem 3, you can print out your counter each time the loop increments.
- Remember to convert Int32 data types to strings before printing them out on the screen.

For More Information:
- [http://msdn.microsoft.com/en-us/library/5z06z1kb.aspx](http://msdn.microsoft.com/en-us/library/5z06z1kb.aspx)
Unit 1 Section 10 Pre-Test

Please take this test before the lesson.

Name: _________________________________

Directions: Answer the questions based on the following code example:

```vbscript
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim diceroll As Integer
    Dim counter As Integer
    Do
        diceroll = Int(Rnd() * 120) + 3
        Response.Write(Convert.ToString(diceroll) & vbCrLf)
        counter = counter + 1
    Loop While counter <> 20
    Response.Write("Done")
End Sub
</script>
```

1. When will ‘Done’ print out?
   a. After the loop runs 20 times
   b. After the loop runs 1 time
   c. When diceroll is equal to 20
   d. When diceroll is not equal to 20

2. What is the highest value of diceroll?
   a. 500
   b. 250
   c. 123
   d. 3

3. What is the lowest value of diceroll?
   a. 122
   b. 123
   c. 3
   d. 4

4. By how much is counter incremented each time the loop runs?
   a. Subtract 1
   b. Add 1
   c. Add 2
   d. Subtract 2

5. What would happen if loop while was replaced with loop until?
   a. The code would run once.
   b. The code would run forever.
   c. 20 would be printed out multiple times
Unit 1 Section 10 Post-Test

Please take this test after the lesson. Answers

Name: __________________________________________

Directions: Answer the questions based on the following code example:

```vbscript
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim diceroll As Integer
    Dim counter As Integer

    Do
        diceroll = Int(Rnd() * 120) + 3
        Response.Write(Converter.ToString(diceroll) & vbCrLf)
        counter = counter + 1
    Loop While counter <> 20
    Response.Write("Done")
End Sub
```

1. When will ‘Done’ print out?
   a. After the loop runs 20 times
   b. After the loop runs 1 time
   c. When diceroll is equal to 20
   d. When diceroll is not equal to 20

2. What is the highest value of diceroll?
   a. 500
   b. 250
   c. 123
   d. 3

3. What is the lowest value of diceroll?
   a. 122
   b. 123
   c. 3
   d. 4

4. By how much is counter incremented each time the loop runs?
   a. Subtract 1
   b. Add 1
   c. Add 2
   d. Subtract 2

5. What would happen if loop while was replaced with loop until?
   a. The code would run once.
   b. The code would run forever.
   c. 20 would be printed out multiple times
Unit 2 Section 1: SQLSelect1
Author: Hadiya
Date Created: 04-26-2014

Please take the Pre-Test before starting this lesson.

Purpose: The purpose of this lesson is to introduce you to SQL.

Overview:

SQL
SQL is an acronym for Structured Query Language. You can use SQL to interact with databases.

Database
A database is a collection of data. This data can be inserted, updated, deleted, and displayed (selected).
In this lesson, we will focus on selecting data.

For this handbook, we will use the MySQL database. This type of database is relational. For more information about relational databases, see this link: http://computer.howstuffworks.com/question599.htm

Select Statements
In this lesson, we will focus on basic select statements.

Please read the following webpages:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Webpage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Introduction</td>
<td><a href="http://www.w3schools.com/sql/sql_intro.asp">http://www.w3schools.com/sql/sql_intro.asp</a></td>
</tr>
<tr>
<td>SQL Syntax</td>
<td><a href="http://www.w3schools.com/sql/sql_syntax.asp">http://www.w3schools.com/sql/sql_syntax.asp</a></td>
</tr>
<tr>
<td>SQL Select</td>
<td><a href="http://www.w3schools.com/sql/sql_select.asp">http://www.w3schools.com/sql/sql_select.asp</a></td>
</tr>
</tbody>
</table>

*Do the try it yourself sections

My SQL Workbench
Please see the Downloads For This Course section if you have not yet downloaded My SQL Workbench, the MySQL ODBC driver, and MySQL Community Server.

In this section, we will teach you how to get into the database, create schemas, tables, and put data into your table.

1. Open MySQL Workbench. The icon looks like this:
2. Open the local instance of MySQL

3. Type in the password that you created when installing the MySQL Community server. Then click ‘OK’.

However, if you see the following, click connect in order to get to the screen pictured above.
4. You now see the layout of MySQL Workbench. Workbench allows you to work with the databases that you create. There are some test databases in the schemas section highlighted below. Click on the test schema. A schema is defined here (http://stackoverflow.com/questions/2674222/what-is-purpose-of-database-schema) as “a way to logically group objects such as tables, views, stored procedures etc.”

5. We will now create a table in the test schema to organize the data. A common table that is found in databases is security. This table will include an ID, firstname, lastname, username, and password. Right click on tables to create a table.

6. Fill in the following values to create the table:
Notice that the columns in databases have data types. Common Data Types include INT and VARCHAR. A VARCHAR data type is like a string. The number in parenthesis after the data type is the number of characters that can be stored each record. Therefore, VARCHAR(45) means that only 45 characters (letters, spaces, numbers etc.) can be used to input into this column. INT(11) means that there can only be 11 digits in each number.

Also, there are many checkboxes next to each column. They stand for Primary Key (PK), Not Null (NN), Unique (UQ), Binary (BIN), Unsigned (UN), Zero Fill (ZF), and Auto Increment (AI).

**Important definitions:**

A **primary key** is a column of a table that makes each record in the table original. Therefore, each record in a table can be referred to by its primary key. In this table, the primary key is ‘idsecurity’. A column such as ‘firstname’ cannot be used as a primary key because two people may have the same first name. Therefore, the record would not be unique.

**Not null** means that the specified column cannot be left empty. The ‘idsecurity’ is not null because this is required for a primary key (see definition of a primary key). Also, username and password are not null because each person needs a username and password.

**Unique** means that each record in the column must be unique. No duplicates can exist. In this table, ‘idsecurity’ is unique because this is required for a primary key (see definition of a primary key). A column like ‘firstname’ is not marked as unique because two people may have the same first name.

**Auto Increment** is a helpful tool that creates a unique number for each record, incrementing by 1 each time. The ‘idsecurity’ is set to auto increment so that each user is automatically assigned an id. A column like ‘firstname’ is not set to autoincrement so that the user can type in their own name.

1. Click ‘Apply’. Then click ‘Apply’ and ‘Finish’ in the pop up window to run the sql that is seen in this popup window.
2. To view your newly created table, find the table in the list of tables. Right click and select ‘Select Rows Limit 1000’.
9. You will now see the screen below. Next, enter data into your table by clicking the edit button. Click on individual cells to add data. You do not need to add anything to the idsecurity column. When you are done, and click apply, the numbers will be automatically entered.

Notice the select statement at the top of the page. This sql was run in order to get the result that you are looking at. The statement should look familiar. It is similar to the statements that you read about earlier in the ww3schools tutorial. However, this statement uses the format schema.table to refer to the table. The schema is test, and the table is security.

You can edit this query in the textbox and run it by clicking the lightning button.
Individual Problems:
Create these tables on your own. See the Tips and Overview section if you need help.

Problem 1: Create a table of library books. Call the table ‘librarybooks’. The columns are:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>PK</th>
<th>NN</th>
<th>UQ</th>
<th>AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookid</td>
<td>INT(11)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>bookname</td>
<td>VARCHAR(45)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bookauthor</td>
<td>VARCHAR(45)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yearpublished</td>
<td>INT(4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add some sample data.

Problem 2: Create a table of classes. Call the table ‘classes’. The columns are:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>PK</th>
<th>NN</th>
<th>UQ</th>
<th>AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>classid</td>
<td>INT(11)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>classname</td>
<td>VARCHAR(45)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>textbook</td>
<td>VARCHAR(45)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notes</td>
<td>VARCHAR(200)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add in some sample data.

Problem 3: Create a twitter table. Call the table ‘twitter’. The columns are:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>PK</th>
<th>NN</th>
<th>UQ</th>
<th>AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>tweetid</td>
<td>INT(11)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>userid</td>
<td>INT(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tweet</td>
<td>VARCHAR(150)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add in some sample data. Make sure that the userids that you use are also found in the idsecurity column of your security table. This is an example of a foreign key. A foreign key is a relationship between one column in one table to another column in another table. In this example, each record in the security table is a user. Each user can compose tweets. Each tweet is a record in the twitter table. Therefore, if a user tweets something, their userid will be recorded in the twitter table so that we can determine who composed the tweet.

Tips:
- When creating a table, do not include spaces in the table name or column names. Consider using the underscore character or dashes if needed.
- Besides right clicking on the table, you can call up each of your tables by writing the select statement.
For More Information:

- [http://computer.howstuffworks.com/question599.htm](http://computer.howstuffworks.com/question599.htm)
- [http://www.w3schools.com/sql/sql_intro.asp](http://www.w3schools.com/sql/sql_intro.asp)
Unit 2 Section 1 Pre-Test

Please take this test before the lesson.

Name: ________________________________

Select a multiple choice letter to answer each question.

1. You create the column ‘lastname’ in the table ‘customers’ to store the last names of customers for your new business. Which of the following should you select for this column of the table?
   a. Auto Increment
   b. Unique
   c. Primary Key
   d. None of the Above

2. In the same table, you want to collect the addresses of all of your customers. However, some customers do not want to provide their addresses. Select a true statement about this column from the given information.
   a. This column should be Not Null.
   b. This column should not be Not Null.
   c. This column should be a Foreign Key.
   d. This column should not be a Foreign Key.

3. You also want to assign each customer a status in the status column. Each customer is labelled ‘ACTIVE’ or ‘INACTIVE’ in the status column. What data type should you use?
   a. INT(11)
   b. INT(40)
   c. VARCHAR(3)
   d. VARCHAR(10)

4. The ‘customers’ table needs a primary key. Which of the following columns can be used as an appropriate primary key?
   a. socialsecuritynumber
   b. status
   c. firstname
   d. lastname

5. Each customer’s purchases are listed in the ‘purchases’ table. The ‘purchases’ table needs a primary key. Which of these columns would be an appropriate primary key?
   a. purchase_id
   b. socialsecuritynumber
   c. itemname
   d. datepurchased
Unit 2 Section 1 Post-Test

Please take this test after the lesson. Answers

Name: ________________________________

Select a multiple choice letter to answer each question.

1. You create the column ‘lastname’ in the table ‘customers’ to store the last names of customers for your new business. Which of the following should you select for this column of the table?
   a. Auto Increment
   b. Unique
   c. Primary Key
   d. None of the Above

2. In the same table, you want to collect the addresses of all of your customers. However, some customers do not want to provide their addresses. Select a true statement about this column from the given information.
   a. This column should be Not Null.
   b. This column should not be Not Null.
   c. This column should be a Foreign Key.
   d. This column should not be a Foreign Key.

3. You also want to assign each customer a status in the status column. Each customer is labelled ‘ACTIVE’ or ‘INACTIVE’ in the status column. What data type should you use?
   a. INT(11)
   b. INT(40)
   c. VARCHAR(3)
   d. VARCHAR(10)

4. The ‘customers’ table needs a primary key. Which of the following columns can be used as an appropriate primary key?
   a. socialsecuritynumber
   b. status
   c. firstname
   d. lastname

5. Each customer’s purchases are listed in the ‘purchases’ table. The ‘purchases’ table needs a primary key. Which of these columns would be an appropriate primary key?
   a. purchase_id
   b. socialsecuritynumber
   c. itemname
   d. datepurchased
Unit 2 Section 2: SQL Select 2
Author: Hadiya
Date Created: 04-26-2014

Please take the Pre-Test before starting this lesson.

Purpose: In Unit 2 Section 1, you learned about databases, and how to display data in a database. The purpose of this lesson (Unit 2 Section 2) is to teach you how to display data in a web application.

Overview:
To display data in a web application, you need to create a DSN. Then, you use two objects: a Gridview and a SQL Datasource. The Gridview and SQL Datasource connect together to display data. This makes your webpage dynamic and more complex because the data in the database may change, but no additional code needs to be written to display the data on the webpage.

Creating a DSN
For each schema that you use, a DSN must be created. DSN stands for Data Source Name. For this lesson, you only need to create one DSN for this lesson because the tables that you created in the previous lesson are all in one schema: ‘test’. Follow the steps below to create a DSN.

1. Go to start and type in Data Sources. Select the 32-bit version. (This step may look different on your computer because this lesson was created in Windows 8.)
2. Go to ‘System DSN’ and click ‘Add’.

3. Select the ‘MySQL ODBC 5.1 Driver’ then click ‘Finish’. If this driver isn’t there, then see the downloads for this course page for instructions about how to download it.

4. Fill in the following information:

   The data source name is the name that you want to refer to this schema as in websites and applications. We recommend that you name the DSN the same as the schema that you are using to avoid confusion.

   The TCP/IP server is the server where the database is located. Because the database is located on your local machine, type in the word ‘localhost’. You can connect to other servers by typing in the IP Address of that server.

   The username and password is the same that was used to login to MySQL. The username is usually root for the local instance of the database.

   Next, the database is the name of the database that you want to make a DSN for. In this case, we are making a DSN for the ‘test’ database.

   Lastly, click the ‘Test’ button to make sure that you can connect to the database.
If the connection is successful, click OK. Then click OK again to get out of ODBC datasources.

Displaying Data Using a Gridview and SQL Datasource

Follow the following steps:

1. Create a new website. We recommend that you call this website ‘basicgridview’.
2. Create a new webpage without code behind. We recommend that you call this page ‘gridsecurity.aspx’.
3. Drag in a Gridview and SQL Datasource from the toolbox. These objects are located under the data tab.
4. Configure the datasource by selecting the data source and clicking the arrow next to it.

5. Then click ‘Configure Data Source’

6. Select New Connection. In the future, once you have set up the connection once, you can select your connection string from the dropdown list to bypass some of the following steps.

7. Select Microsoft ODBC Data Sources if it is not already selected.
8. Enter the system DSN, username, and password that you normally use for the database. Then click ‘Test Connection’. If the connection succeeds, then click ‘OK’.
9. Click Next.
10. If you want to use this connection information in the future keep the box checked and click next. This step is recommended. Then click next.

11. Keep clicking ‘Next’. These steps are not applicable to the type of database that we use. Then click ‘Finish’.
12. Go to ‘Source’ mode and find the SQL Data Source in the HTML code.
13. One of the properties is ‘selectcommand’. The select command as it appears below is wrong due to the brackets.

```html
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="%$ ConnectionStrings:ConnectionString2 %$"
    ProviderName="%$ ConnectionStrings:ConnectionString2.ProviderName %$"
    SelectCommand="SELECT * FROM [security]""></asp:SqlDataSource>
```

Replace the select command with the correct select command selecting the security table (see the Overview section from Unit 2 Lesson 1 for more information). The correct select command is “Select * From security”.

```html
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="%$ ConnectionStrings:ConnectionString2 %$"
    ProviderName="%$ ConnectionStrings:ConnectionString2.ProviderName %$"
    SelectCommand="SELECT * FROM security"></asp:SqlDataSource>
```

14. Connect the datasource to the gridview. Do this by going to ‘Design’ mode. Then click the arrow on the side of the gridview. Select the correct data source from the dropdown list.
15. Autoformat the gridview. This function is found above ‘Choose Data Source’ in the menu that appears after you click the arrow near the gridview. The autoformat function provides many color templates for your gridview. Choose a template.

16. At the top of the page, write import name space statements. These input libraries of code that you can use. These statements should be put at the top of every page that connects to the database. However, they can be put on any page. Get into the habit of including these input namespace statements at the top of every page.

```
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.Odbc" %>
```

17. Debug. This is what appears when I debug my page:

```
id    security    firstname    lastname    username    password
1      123         hhh          hhh          hhh          hhh
2      123         abc          doremi       abc
3      123         hello        world        life         awesome
```
Individual Problems:
Create separate pages for each of these problems. See the Tips section for more help.

Problem 1: Display the ‘library’ table.

Problem 2: Display the ‘classes’ table. Display the ‘classid’ and ‘classname’ columns only.

Problem 3: Display the ‘twitter’ table. Display all of the columns except the ‘userid’.

Tips:
- Remember your import namespace statements (see the Overview for more information).
- Remember to connect your gridview to its corresponding data source.
- Remember to configure all datasources.

For More Information:
Unit 2 Section 2 Pre-Test

Please take this test before the lesson.

Name: __________________________________________

Please select a multiple choice answer.

1. Which two objects do you need to display data from a database?
   a. Gridview, Dataconnector
   b. Dataconnector, Dataviewer
   c. Dataviewer, SQL Data Source
   d. Gridview, SQL Data Source

2. What does DSN stand for?
   a. Dictionary System Network
   b. Direct Series Network
   c. Data Source Name
   d. Data Sequence Name

3. selectcommand is a property of which object?
   a. Gridview
   b. SQL Data Source
   c. Dataconnector
   d. Dataviewer

4. What does the autoformat function do?
   a. Change the colors of the object
   b. Change the object type
   c. Change the data type
   d. Change the name of the object

5. How many DSN’s must be created per schema?
   a. 1
   b. 2
   c. 3
   d. 4

6. Which driver should you use to connect MySQL to Visual Web Developer?
   a. VWD-SQL 3.0 Driver
   b. MySQL ODBC 5.1 Driver
   c. MyVWD 8.0 Driver

7. Choose the correct select statement to select data from the security table:
   a. SELECT * FROM [security];
   b. SELECT * FROM (security);
   c. SELECT * FROM security;
   d. All of the above
Unit 1 Section 2 Post-Test

Please take this test after the lesson. Answers

Name:_________________________________

Please select a multiple choice answer.

1. Which two objects do you need to display data from a database?
   a. Gridview, Dataconnector
   b. Dataconnector, Dataviewer
   c. Dataviewer, SQL Data Source
   d. Gridview, SQL Data Source

2. What does DSN stand for?
   a. Dictionary System Network
   b. Direct Series Network
   c. Data Source Name
   d. Data Sequence Name

3. selectcommand is a property of which object?
   a. Gridview
   b. SQL Data Source
   c. Dataconnector
   d. Dataviewer

4. What does the autoformat function do?
   a. Change the colors of the object
   b. Change the object type
   c. Change the data type
   d. Change the name of the object

5. How many DSN’s must be created per schema?
   a. 1
   b. 2
   c. 3
   d. 4

6. Which driver should you use to connect MySQL to Visual Web Developer?
   a. VWD-SQL 3.0 Driver
   b. MySQL ODBC 5.1 Driver
   c. MyVWD 8.0 Driver

7. Choose the correct select statement to select data from the security table:
   a. SELECT * FROM [security];
   b. SELECT * FROM (security);
   c. SELECT * FROM security;
   d. All of the above
Unit 2 Section 3: SQL Select 3
Author: Hadiya
Date Created: 04-26-2014

Please take the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to introduce you to advanced select statements.

Overview:
Please read all of the information below, and do the try it sections in the ww3 schools lessons below.

<table>
<thead>
<tr>
<th>Lesson Topic</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Distinct</td>
<td><a href="http://www.w3schools.com/sql/sql_distinct.asp">http://www.w3schools.com/sql/sql_distinct.asp</a></td>
</tr>
<tr>
<td>Where</td>
<td><a href="http://www.w3schools.com/sql/sql_where.asp">http://www.w3schools.com/sql/sql_where.asp</a></td>
</tr>
<tr>
<td>And &amp; Or</td>
<td><a href="http://www.w3schools.com/sql/sql_and_or.asp">http://www.w3schools.com/sql/sql_and_or.asp</a></td>
</tr>
<tr>
<td>Order By</td>
<td><a href="http://www.w3schools.com/sql/sql_orderby.asp">http://www.w3schools.com/sql/sql_orderby.asp</a></td>
</tr>
<tr>
<td>Select Top</td>
<td><a href="http://www.w3schools.com/sql/sql_top.asp">http://www.w3schools.com/sql/sql_top.asp</a></td>
</tr>
<tr>
<td>Like</td>
<td><a href="http://www.w3schools.com/sql/sql_like.asp">http://www.w3schools.com/sql/sql_like.asp</a></td>
</tr>
<tr>
<td>Wildcards</td>
<td><a href="http://www.w3schools.com/sql/sql_wildcards.asp">http://www.w3schools.com/sql/sql_wildcards.asp</a></td>
</tr>
<tr>
<td>In</td>
<td><a href="http://www.w3schools.com/sql/sql_in.asp">http://www.w3schools.com/sql/sql_in.asp</a></td>
</tr>
<tr>
<td>Between</td>
<td><a href="http://www.w3schools.com/sql/sql_between.asp">http://www.w3schools.com/sql/sql_between.asp</a></td>
</tr>
<tr>
<td>Functions</td>
<td><a href="http://www.w3schools.com/sql/sql_functions.asp">http://www.w3schools.com/sql/sql_functions.asp</a></td>
</tr>
<tr>
<td>Average</td>
<td><a href="http://www.w3schools.com/sql/sql_func_count.asp">http://www.w3schools.com/sql/sql_func_count.asp</a></td>
</tr>
<tr>
<td>Count</td>
<td><a href="http://www.w3schools.com/sql/sql_func_count.asp">http://www.w3schools.com/sql/sql_func_count.asp</a></td>
</tr>
<tr>
<td>Max</td>
<td><a href="http://www.w3schools.com/sql/sql_func_max.asp">http://www.w3schools.com/sql/sql_func_max.asp</a></td>
</tr>
<tr>
<td>Min</td>
<td><a href="http://www.w3schools.com/sql/sql_func_min.asp">http://www.w3schools.com/sql/sql_func_min.asp</a></td>
</tr>
<tr>
<td>Sum</td>
<td><a href="http://www.w3schools.com/sql/sql_func_sum.asp">http://www.w3schools.com/sql/sql_func_sum.asp</a></td>
</tr>
<tr>
<td>Group By</td>
<td><a href="http://www.w3schools.com/sql/sql_groupby.asp">http://www.w3schools.com/sql/sql_groupby.asp</a></td>
</tr>
<tr>
<td>Having</td>
<td><a href="http://www.w3schools.com/sql/sql_having.asp">http://www.w3schools.com/sql/sql_having.asp</a></td>
</tr>
<tr>
<td>Uppercase</td>
<td><a href="http://www.w3schools.com/sql/sql_func_ucase.asp">http://www.w3schools.com/sql/sql_func_ucase.asp</a></td>
</tr>
<tr>
<td>Lowercase</td>
<td><a href="http://www.w3schools.com/sql/sql_func_lcase.asp">http://www.w3schools.com/sql/sql_func_lcase.asp</a></td>
</tr>
<tr>
<td>Round</td>
<td><a href="http://www.w3schools.com/sql/sql_func_round.asp">http://www.w3schools.com/sql/sql_func_round.asp</a></td>
</tr>
<tr>
<td>Now</td>
<td><a href="http://www.w3schools.com/sql/sql_func_now.asp">http://www.w3schools.com/sql/sql_func_now.asp</a></td>
</tr>
</tbody>
</table>

Gridview- Enable Sorting and Paging
To enable sorting and paging, click the arrow on the side of the gridview. Then check the appropriate boxes.
Rename columns using SQL
You can keep a column name the same in a database, but display another name using SQL. To do this, use the word as. For example,

Select count(userid) as 'Number of Users', max(userid) as 'Last Created User ID'

Make sure to put single quotes '' around the blocks of text.

Problem Introduction:
Problem 1: Display the ‘userid’ and ‘tweet’ columns of the twitter table. Order the records by userid. Make the tweets all lowercase. Enable paging and sorting.

Problem 2: Display the ‘userid’ and ‘tweet’ columns of the twitter table. Select the records who were created by the ‘userid’ 1 or has a ‘tweetid’ of 1. Rename ‘userid’ to ‘user’ in the gridview. Make the tweets all uppercase. Enable paging and sorting.

Problem 3: Perform the following actions, all in one gridview. Use the ‘security’ table. Select the maximum ‘idsecurity’, minimum ‘idsecurity’, average ‘idsecurity’, number of records in the table, round the sum of the ‘idsecurity’ column divided by 2, and the current date and time. Enable paging and sorting.

Problem 4: Select from the ‘librarybooks’ table any ‘bookname’ that has an ‘a’ or ‘b’ in it. Enable paging and sorting.

Problem 5: Select from the ‘classes’ table any classes that have a ‘classid’ in between 2 and 4. Enable paging and sorting.

Problem 6: Select from the ‘classes’ table any classes that have a ‘classid’ less than 4. Enable paging and sorting.

Individual Problems:
Create a new website. We recommend that you call the website ‘gridviewadvanced’.

All of the problems above (1-6) are individual problems. Read the Overview and check the Tips section if you need any help. Also, Unit 2 Lesson 1 and Unit 2 Lesson 2 are very helpful. One of the harder SQL Statements is provided below.

Selected SQL Statements:
Problem 3: SELECT max(idsecurity), min(idsecurity), avg(idsecurity), count(idsecurity), round(sum(idsecurity)/2), now() FROM test.security;

Tips:
- You can test your queries in MySQL Workbench before using them in a website.
  - If you do this, remember to refer to columns in the format column.schema
  - Or right click the schema and set it as the default schema.
- Remember the two import namespace statements.
• Remember to connect your Gridview to the correct SQLDataSource.
• Do not use brackets around the column names or table names in your SQL statements.
• The select statements should be placed in the SQLDataSource property called select command. (See Unit 2 Section 2 for more information)

For More Information:
• http://www.w3schools.com/sql/default.asp
Unit 2 Section 3 Pre-Test

Please take this test before the lesson.

Name: ____________________________________________

Use the table below to answer the questions.

Table Name: grocerystore
Select Statement: SELECT * FROM grocerystore;

<table>
<thead>
<tr>
<th>itemid</th>
<th>itemname</th>
<th>numberleft</th>
<th>aisle</th>
<th>supplierid</th>
<th>datedelivered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Grapes</td>
<td>20</td>
<td>1</td>
<td>34</td>
<td>05-14-13</td>
</tr>
<tr>
<td>2</td>
<td>t-shirts</td>
<td>30</td>
<td>4</td>
<td>78</td>
<td>01-01-01</td>
</tr>
<tr>
<td>3</td>
<td>Blue jeans</td>
<td>10</td>
<td>10</td>
<td>91</td>
<td>02-24-15</td>
</tr>
<tr>
<td>4</td>
<td>Tortilla Chips</td>
<td>20</td>
<td>6</td>
<td>30</td>
<td>08-09-10</td>
</tr>
<tr>
<td>5</td>
<td>Juice</td>
<td>15</td>
<td>9</td>
<td>230</td>
<td>10-10-14</td>
</tr>
<tr>
<td>6</td>
<td>Lettuce</td>
<td>25</td>
<td>2</td>
<td>12</td>
<td>03-21-14</td>
</tr>
</tbody>
</table>

1. Which ‘itemid’ would be selected if the following select statement was executed?
   SELECT * FROM grocerystore where numberleft=25;
   a. Lettuce
   b. 6
   c. 2
   d. 230

2. Which number would appear if the following select statement was executed?
   SELECT count(*) FROM grocery store;
   a. 6
   b. 5
   c. 4
   d. 3

3. Which number would appear if the following select statement was executed?
   SELECT sum(aisle)/2 FROM grocerystore WHERE itemname='Grapes';
   a. 1
   b. .5
   c. 32
   d. 16

4. Which ‘itemid’ numbers would appear if the following select statement was executed?
   SELECT * FROM grocerystore WHERE aisle<6;
   a. 1,2,3,4,5,6
   b. 1,2,3,4,5
   c. 1,3,5
   d. 1,2,6
Unit 2 Section 3 Post-Test
Please take this test after the lesson. Answers

Name: ________________________________

Use the table below to answer the questions.

Table Name: grocerystore
Select Statement: SELECT * FROM grocerystore;

<table>
<thead>
<tr>
<th>Itemid</th>
<th>Itemname</th>
<th>numberleft</th>
<th>Aisle</th>
<th>supplierid</th>
<th>datedelivered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Grapes</td>
<td>20</td>
<td>1</td>
<td>34</td>
<td>05-14-13</td>
</tr>
<tr>
<td>2</td>
<td>T-shirts</td>
<td>30</td>
<td>4</td>
<td>78</td>
<td>01-01-01</td>
</tr>
<tr>
<td>3</td>
<td>Blue jeans</td>
<td>10</td>
<td>10</td>
<td>91</td>
<td>02-24-15</td>
</tr>
<tr>
<td>4</td>
<td>Tortilla Chips</td>
<td>20</td>
<td>6</td>
<td>30</td>
<td>08-09-10</td>
</tr>
<tr>
<td>5</td>
<td>Juice</td>
<td>15</td>
<td>9</td>
<td>230</td>
<td>10-10-14</td>
</tr>
<tr>
<td>6</td>
<td>Lettuce</td>
<td>25</td>
<td>2</td>
<td>12</td>
<td>03-21-14</td>
</tr>
</tbody>
</table>

1. Which ‘itemid’ would be selected if the following select statement was executed?
   SELECT * FROM grocerystore where numberleft=25;
   a. Lettuce  
   b. 6  
   c. 2  
   d. 230

2. Which number would appear if the following select statement was executed?
   SELECT count(*) FROM grocery store;
   a. 6  
   b. 5  
   c. 4  
   d. 3

3. Which number would appear if the following select statement was executed?
   SELECT sum(aisle)/2 FROM grocerystore WHERE itemid='grapes';
   a. 1  
   b. .5  
   c. 32  
   d. 16

4. Which ‘itemid’ numbers would appear if the following select statement was executed?
   SELECT * FROM grocerystore WHERE aisle<6;
   a. 1,2,3,4,5,6  
   b. 1,2,3,4,5  
   c. 1,3,5  
   d. 1,2,6
Unit 2 Section 4: Logins

Author: Hadiya
Date Created: 05-02-2014

Please complete the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to teach you how to make a database-based login. Also, session variables and documentation will be introduced.

Overview:

Documentation
When programming, it is often not necessary to completely write code yourself. There are plenty of examples that you can copy and paste code from. Then, you can edit parts of the code for your specific needs.

We will use Microsoft’s Documentation to create a login.

To access Documentation, go to the ‘Help’ menu and click ‘View Help’. This will most likely direct you to an online version of the documentation. However, offline versions do exist.

When looking for code examples, make sure that the VB tab is selected.

Examples

The following code example uses a Login control to provide a UI for logging in to a Web site.

Look up the Login Class:

Open Documentation. In the search bar, type Login Class. This search query will allow you to look at the properties, methods, and more for the Login Object. More importantly, there are code examples at the bottom of the page.
Logins
A login is a tool used to authenticate users for a system. Most commonly, a user must supply a username and password to prove that they are a real user. For many types of logins the login object can be used. The following tutorial will teach you how create a login using the login object.

1. Create a new website. We recommend that you call this website ‘logins’.
2. Create two new webpages without code behind. We recommend that you call the first webpage ‘login.aspx’ and the second webpage ‘home.aspx’. The second page is a page that the login will redirect to if the login is successful.
3. On the top of both pages, insert your import namespace statements.

```vbnet
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.Odbc" %>
```

4. Go to the ‘login.aspx’ page. Drag in a login object from the toolbox. This is located under the login tab.

5. Auto Format your login by clicking on the arrow to the right of the login while the login is selected. Rename your login (ID property) if you would like to.
6. Double click your Login object to create a subroutine (in the script) that will run when the login authenticates. Therefore, you will notice that the property ‘onauthenticate’ will be set to the name of the new subroutine that you created.

```vbscript
Protected Sub Login1_Authenticate(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)
    End Sub
</script>
```

7. Now, we need to get the code for this subroutine. Therefore, we will use Documentation.
8. Open documentation and search for the ‘Login Class’.
9. Find the ‘onauthenticate’ method and open it.

| OnAuthenticate | Raises the Authenticate event to authenticate the user. |

10. Go to the examples. Copy this code into the onauthenticate subroutine that you just created.

```vbscript
Overloads Sub OnAuthenticate(ByVal e As AuthenticateEventArgs)
    Dim Authenticated As Boolean
    Authenticated = SiteSpecificAuthenticationMethod(Username, Password)
    e.Authenticated = Authenticated
End Sub
```

In Visual Web Developer:

```vbscript
Protected Sub Login1_Authenticate(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)
    Dim Authenticated As Boolean
    Authenticated = SiteSpecificAuthenticationMethod(Username, Password)
    e.Authenticated = Authenticated
End Sub
```

11. Change the username to the LoginID.Username. Change Password to LoginID.Password. For example, if the id of your login is Login1:
12. Then, copy this code outside of the onauthenticate subroutine.

```vbnet
Protected Sub Login1_Authenticate(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)
    Dim Authenticated As Boolean
    Authenticated = SiteSpecificAuthenticationMethod(Login1.UserName, Login1.Password)
    e.Authenticated = Authenticated
End Sub

Private Function SiteSpecificAuthenticationMethod(ByVal UserName As String, ByVal Password As String) As Boolean
    ' Insert code that implements a site-specific custom authentication method here.
    ' This example implementation always returns false.
    Return False
End Function
```

In Visual Web Developer:

```html
<script runat="server">
Protected Sub Login1_Authenticate(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)
    Dim Authenticated As Boolean
    Authenticated = SiteSpecificAuthenticationMethod(Login1.UserName, Login1.Password)
    e.Authenticated = Authenticated
End Sub

Private Function SiteSpecificAuthenticationMethod(ByVal UserName As String, ByVal Password As String) As Boolean
    ' Insert code that implements a site-specific custom authentication method here.
    ' This example implementation always returns false.
    Return False
End Function
</script>
```

13. Delete from the first comment to Return False.

Before Deletion:

```html
<script runat="server">
Protected Sub Login1_Authenticate(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)
    Dim Authenticated As Boolean
    Authenticated = SiteSpecificAuthenticationMethod(Login1.UserName, Login1.Password)
    e.Authenticated = Authenticated
End Sub

Private Function SiteSpecificAuthenticationMethod(ByVal UserName As String, ByVal Password As String) As Boolean
    ' Insert code that implements a site-specific custom authentication method here.
    ' This example implementation always returns false.
    Return False
End Function
</script>
```

After Deletion:

```vbnet
Private Function SiteSpecificAuthenticationMethod(ByVal UserName As String, ByVal Password As String) As Boolean

End Function
```
14. Open documentation and search ‘odbcdatareader’. Make sure that this is all one word.

```
@odbcdatareader
```

15. Find the code examples at the bottom of the page. Make sure that you are in the ‘VB’ tab.

```
**Examples**

The following example creates an OdbcConnection, an OdbcCommand, and an OdbcDataReader. The example reads through the data, writing it out to the console. Finally, the example closes the OdbcDataReader, and then the OdbcConnection.

```vbnet
Public Sub ReadData(ByVal connectionString As String)
    Dim queryString As String = "SELECT DISTINCT CustomerID FROM Orders"
    Using connection As New OdbcConnection(connectionString)
        Dim command As New OdbcCommand(queryString, connection)
        connection.Open()
        Dim reader As OdbcDataReader = command.ExecuteReader()
        While reader.Read()
            Console.WriteLine("CustomerID={0}", reader(0).ToString)
        End While
        ' Call Close when done reading.
        reader.Close()
    End Using
End Sub
```
```

16. Copy the following code: (From ‘Dim’ to ‘End Using’)

```
**Examples**

The following example creates an OdbcConnection, an OdbcCommand, and an OdbcDataReader. The example reads through the data, writing it out to the console. Finally, the example closes the OdbcDataReader, and then the OdbcConnection.

```vbnet
Public Sub ReadData(ByVal connectionString As String)
    Dim queryString As String = "SELECT DISTINCT CustomerID FROM Orders"
    Using connection As New OdbcConnection(connectionString)
        Dim command As New OdbcCommand(queryString, connection)
        connection.Open()
        Dim reader As OdbcDataReader = command.ExecuteReader()
        While reader.Read()
            Console.WriteLine("CustomerID={0}", reader(0).ToString)
        End While
        ' Call Close when done reading.
        reader.Close()
    End Using
End Sub
```
```

This code connects to the database, selects data in the database, and allows you to work with the code.
17. Paste the code that you copied into the function (not the subroutine).

```vbnet
Private Function SiteSpecificAuthenticationMethod(ByVal UserName As String, ByVal Password As String) As Boolean
    Dim queryString As String = "SELECT DISTINCT CustomerID FROM Orders"

    Using connection As New OdbcConnection(connectionString)
        Dim command As New OdbcCommand(queryString, connection)
        connection.Open()
        Dim reader As OdbcDataReader = command.ExecuteReader()
        While reader.Read()
            Console.WriteLine("CustomerID={0}", reader(0).ToString)
        End While
    End Using
End Function
```

18. Now we will edit the code to work for our Login.
   - The query string is wrong. Set the ‘queryString’ variable equal to a select statement for the security table.

```vbnet
Dim queryString As String = "SELECT * FROM security"
```

   - There is no ‘connectionString’. Create a new string variable called ‘connectionString’ and set it equal to this string “dsn=putdsnhere;uid=putuseridhere;pwd=putpasswordhere”

```vbnet
Dim connectionString As String = "dsn=test;uid=root;pwd=password"
```

   - There is no code authenticating the login.

Take out the following line of code in the While loop.

```vbnet
    While reader.Read()
        Console.WriteLine("CustomerID={0}", reader(0).ToString)
    End While
    ' Call Close when done reading.
    reader.Close()
End Using
```

This while loop runs while the database is being read. In this loop, we can do database operations, replacing the line that we just deleted. In this case we will use if statements to verify the username and passwords:
While reader.Read()
    If Login1.UserName = reader(3) Then
        If Login1.Password = reader(4) Then
            Return True
        End If
    End If
End While
Return False

Notes:

- Username is a property of a Login object. Therefore, Login1.UserName refers to the username textbox in the Login object.
- Password is a property of a Login object. Therefore, Login1.Password refers to the username textbox in the Login object.
- Reader(#) is a column of a database. Open MySQL Workbench and look at your security table. The columns are referred to by numbers starting at 0.

```
SELECT * FROM test.security;
```

- idsecurity is column 0, so it is referred to as reader(0)
- firstname is column 1, so it is referred to as reader(1)
- lastname is column 2, so it is referred to as reader(2)
- username is column 3, so it is referred to as reader(3)
- password is column 4, so it is referred to as reader(4)

- Return True prompts the Login Object to perform certain actions that authenticate the user.
- Return False prompts the Login Object to perform certain actions that do not authenticate the user.

19. Lastly, we can change the properties of the Login to customize what happens after the ‘Return True’ and ‘Return False’. These properties are ‘FailureText’ and ‘DestinationPageURL’. The failure text will appear if the function returns false. The user will be redirected to the destination page url if the function returns true.
20. To change the Failure Text, simply change the text that already exists as failure text.

21. For this application, we need to change the destination page URL to a page in our website called ‘home.aspx’. To do this, click the ‘…’ button and select the home page.

You can also redirect to pages such as Google by typing in the full URL.

22. Debug. Test correct logins from your database and test incorrect logins.

Session Variables

A session variable is a type of variable that exists on all of the pages in a website. Therefore, you can create the variable on one page and use the information stored in the variable on another page.

When you open your email, you may see a message on the top of the screen greeting you with your name or username. A session variable can be used to create this message.

Creating Session Variables:

The general format for creating a session variable is:

Session ("nameindoublequotes") = “value”
In this problem, we want to create a session variable called “Login” that stores the user’s username. Therefore, on the ‘login.aspx’ page in the while loop, insert the statement ‘Session(“login”)=reader(3)’.

```csharp
While reader.Read()
    If Login1.UserName = reader(3) Then
        If Login1.Password = reader(4) Then
            Session(”login”) = reader(3)
            Return True
        End If
    End If
End While
Return False
```

Note: Do not put ‘Session(“login”)=reader(3)’ after ‘Return True’. If you do, then the function will return true and redirect to another page before the line of code ‘Session(“login”)=reader(3)’ runs.

Using Session Variables:

Use a Session Variable just like you would use a string variable.

For this problem, we want to display ‘Hi Username’ on the home page. Therefore, make a new label on the home page and double click on the page to create a page load subroutine.

```html
<asp:label ID="Label1" runat="server" Text="Label"></asp:label>
```

In the page load subroutine, set the text of the label equal to “Hi ” and the session variable (use concatenation).

```csharp
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Label1.Text = "Hi " & Session("Login")
End Sub
```
Lastly, when you debug, make sure that you start debugging on the login page. If you start debugging on the home page, the session variable will not have any value, and the page will simply say ‘Hi’.

Problem Introduction:

**Problem 1:** Create a specialized login. A school wants to have a login for both students and teachers. Create a new database table, called schoollogin for the problems in this lesson, as detailed below:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>PK</th>
<th>NN</th>
<th>UQ</th>
<th>AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>INT(11)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>firstname</td>
<td>VARCHAR(45)</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastname</td>
<td>VARCHAR(45)</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>role</td>
<td>VARCHAR(45)</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>VARCHAR(45)</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>VARCHAR(45)</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fill in data for the table in the database. The only roles that should exist are ‘Student’ and ‘Teacher’. Make sure that you are consistent in how you input ‘Student’ and ‘Teacher’ into the table. For example, if you put ‘student’ instead of ‘Student’ if you run the sql command SELECT * FROM schoollogin where role='Student’, the record where role is ‘student’ will not show up due to the lowercase ‘s’.

Students should be redirected to the page ‘student.aspx’. Teachers should be redirected to the page ‘teacher.aspx’. On the top of these pages greet the students and teachers by username (use a session variable).

**Problem 2:** Re-create the 2 page basic login with session variable that was created in the overview. However, use your newly created schoollogin table. Try your best to do this from memory! Please use documentation to help you.

**Problem Programming Steps:**

**Problem 1:**

1. Create the schoollogin table. See the chart in Problem 1 for help.
2. Create three new webpages. We recommend that you call the first webpage ‘schoollogin.aspx’. The problem specifies that the other two pages should be called ‘student.aspx’ and ‘teacher.aspx’.
3. Put import namespace commands on each page.

```vb
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.Odbc" %>
```
4. Create a login on the ‘schoollogin.aspx’ page as directed in the overview above. Be sure to use the correct select statement for the ‘schoollogin’ table. Also, make sure that you are using the
correct column number for readers in your while loop. Lastly, create a session variable for the username of the student.

5. Delete Return true on the ‘schoollogin.aspx’ page. This will only allow you to direct potential users to one page.

While reader.Read()
  If Login1.UserName = reader(4) Then
    If Login1.Password = reader(5) Then
      Session("username") = reader(4)
      Return True
    End If
  End If
End While

6. Create two more if statements nested within the other two if statements. If the role is equal to ‘Teacher’ then redirect to the page ‘teacher.aspx’. If the role is equal to ‘Student’ then redirect to the page ‘student.aspx’. If the role is equal to ‘Teacher’ then redirect to the page ‘teacher.aspx’.

  *Make sure that you use the correct reader for the role column
While reader.Read()
    If Login1.UserName = reader(4) Then
        If Login1.Password = reader(5) Then
            Session("username") = reader(4)
            If reader(3) = "Teacher" Then
                Response.Redirect("teacher.aspx")
            End If
            If reader(3) = "Student" Then
                Response.Redirect("student.aspx")
            End If
        End If
    End If
End While

' Call Close when done reading.
reader.Close()
End Using
Return False

7. On the student and teacher pages greet the user by name using a session variable and a label.

student.aspx

teacher.aspx
8. Debug your application. Test correct and incorrect passwords. Also, test both student and teacher user accounts.

**Individual Problems:**

Problem 2 is an individual problem. See the **Tips** section and overview sections if you need help.

**Tips:**

- Remember to put in your import namespace statements.
- If you ever are unable to find the toolbox, go to the ‘Window’ menu and select ‘Reset Window Layout’
- When debugging your website, try to test as many combinations as possible. For example, for problem 1, we asked that you test both student and teacher accounts.
- To memorize how to create a login, I realize that I need to copy the code in, and address all of the errors that I see. For example, to fix the error on step 10, I changed Username to
Login1.Username and Password to Login1.Password. Also, to fix the error on the variable ‘connectionString’, I created a variable called ‘connectionString’

For More Information:
Unit 2 Section 4 Pre-Test

Please take this test before the lesson.

Name: ____________________________________________

Please answer each question by selecting a multiple choice answer.

1. What two keywords should you search for in documentation to create a login?
   a. Login Class, odbcdatareader
   b. Textbox Class, RadioButtonList Class
   c. Textbox Class, Login Class
   d. odbcdatareader, Textbox Class

2. What statements should you add near the top of every page that does database operations?
   a. < %@ Import Namespace="System.Data" %>
   < %@ Import Namespace="System.Data.Odbc" %>
   b. < %@ Import Variables="System.Data" %>
   < %@ Import Variables="System.Data.Odbc" %>
   c. Dim Namespace1 as String="System.Data.Odbc"
      Dim Namespace2 as String="System.Data.Odbc"
   d. Dim Namespace1 as Namespace="System.Data.Odbc"
      Dim Namespace2 as Namespace="System.Data.Odbc"

3. Which statement is true?
   a. A session variable defined on one page can be used on other pages
   b. A string variable defined on one page can be used on other pages
   c. An integer variable defined on one page can be used on other pages
   d. A boolean variable defined on one page can be used on other pages

4. Which subroutine is triggered after the ‘Log In’ button on the Login Object is pressed?
   a. onauthenticate
   b. onlogin
   c. onbuttonpressed
   d. whenheuserlogsin

Look at the following table, and use the data in it to answer the following question.

<table>
<thead>
<tr>
<th>userid</th>
<th>password</th>
<th>security</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>543@43</td>
<td>Administrator</td>
<td>Waffles</td>
</tr>
<tr>
<td>2</td>
<td>1235423</td>
<td>Level 1</td>
<td>h@t</td>
</tr>
<tr>
<td>3</td>
<td>Beta00</td>
<td>Level 2</td>
<td>Brick</td>
</tr>
</tbody>
</table>

5. Which statement is true?
   a. The userid column is reader(1)
   b. The password column is reader(2)
   c. The security column is reader(3)
   d. The username column is reader(3)
Unit 2 Section 4 Post-Test
Please take this test after the lesson. Answers

Name: __________________________________________

Please answer each question by selecting a multiple choice answer.

1. What two keywords should you search for in documentation to create a login?
   a. Login Class, odbcdatareader
   b. Textbox Class, RadioButtonList Class
   c. Textbox Class, Login Class
   d. odbcdatareader, Textbox Class

2. What statements should you add near the top of every page that does database operations?
   a. <%@ Import Namespace="System.Data" %>
      <%@ Import Namespace="System.Data.Odbc" %>
   b. <%@ Import Variables="System.Data" %>
      <%@ Import Variables="System.Data.Odbc" %>
   c. Dim Namespace1 as String="System.Data.Odbc"
      Dim Namespace2 as String= "System.Data.Odbc"
   d. Dim Namespace1 as Namespace="System.Data.Odbc"
      Dim Namespace2 as Namespace="System.Data.Odbc"

3. Which statement is true?
   a. A session variable defined on one page can be used on other pages
   b. A string variable defined on one page can be used on other pages
   c. An integer variable defined on one page can be used on other pages
   d. A boolean variable defined on one page can be used on other pages

4. Which subroutine is triggered after the ‘Log In’ button on the Login Object is pressed?
   a. onauthenticate
   b. onlogin
   c. onbuttonpressed
   d. whentheuserlogsin

Look at the following table, and use the data in it to answer the following question.

<table>
<thead>
<tr>
<th>userid</th>
<th>password</th>
<th>Security</th>
<th>username</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>543@43</td>
<td>Administrator</td>
<td>waffles</td>
</tr>
<tr>
<td>2</td>
<td>1235423</td>
<td>Level 1</td>
<td>h@t</td>
</tr>
<tr>
<td>3</td>
<td>Beta00</td>
<td>Level 2</td>
<td>brick</td>
</tr>
</tbody>
</table>

5. Which statement is true?
   a. The userid column is reader(1)
   b. The password column is reader(2)
   c. The security column is reader(3)
   d. The username column is reader(3)
Unit 2 Section 5: SQL Select 4
Author: Hadiya
Date Created: 05-03-2014

Please complete the Pre-Test before starting this lesson.

Purpose:
The purpose of this lesson is to teach you how to create search functions for gridviews in your application.

Overview:
You previously learned how to display data in a Gridview and use different statements to sort or limit the data that the user sees. This lesson will teach you how to create search functions for the user to use.

We can do this using concatenation with in our select statements.
For example, I may want to use this select command in the sub page load sub routine:

Sqldatasource1.selectcommand= "Select * From schoollogin where firstname='first name from a textbox'"

In order to fill in the ‘first name from a textbox’ segment, I can concatenate the text in the select command and the text in the textbox:

Sqldatasource1.selectcommand= "Select * From schoollogin where firstname='" & Textbox1.Text & "'"
• Do not forget to include the tick marks around the textbox text. These are needed because the firstname column in the database is for text (it is VARCHAR).

If I ran the line of code:
Response.Write(Sqldatasource1.selectcommand)
And Textbox1.Text= "Brian"
The following would appear:
Select * From schoollogin where firstname='Brian’

This can also be done with other objects such as dropdownlists, radiobuttons, labels, and more.
Also, using Response.write to look at your sql statement is a good way of checking if you did your concatenation correctly.
Lastly, make sure that you put these lines of code in the script. A good sub routine to put it in is sub page load.
Problem Introduction:

Problem 1:

A school has requested that you make an application that allows school teachers to login and look at the login table. Use your schoollogin table for this application.

Create a login page called ‘admin.aspx’. Only allow teachers to login. Direct authenticated teachers to the page ‘logintable.aspx’.

On the ‘logintable.aspx’ page, greet the teachers by username (using a session variable).

Below the greeting, display the login table. Allow teachers to search the login table. The teachers can search the table with:

- A drop down list containing the options: ‘Both’, ‘Student’, and ‘Teacher’.
  - The search parameters will be activated when the user clicks the ‘Search by Role’ button.
- Search within a limited range of ids. Use two textboxes: one for the lower id and the other for the higher id.
  - If both textboxes are not filled in, warn the user and do not search the database.
  - The search parameters will be activated when the user clicks the ‘Search by ID’ button.
- Search by first name
  - Use a textbox to collect this information
  - The search parameters will be activated when the user clicks the ‘Search by First Name’ button.
- Search by last name
  - Use a textbox to collect this information
  - The search parameters will be activated when the user clicks the ‘Search by Last Name’ button.
- Reset button
  - Reset the select statement to display everything in the ‘schoollogin’ table.
  - Also, the reset button should clear all of the textboxes.

We recommend that you use a table to organize everything. However, any method of organization may be used.

Problem 2:

Create a twitter feed using the twitter table. Allow the user to search by ‘userid’ using a textbox. Also allow the user to search by text in the tweets using a textbox. For example, the user should be able to find a single word within any tweet. See the tips section for help.
Problem Programming Steps:

Problem 1:

1. Create a new website. We recommend that you call this website 'searchgridview'.
2. Create two new webpages ('admin.aspx' and 'logintable.aspx') as designated by the problem.
3. Put import namespace statements at the top of each page.
4. Create a login that only allows teachers to login. To do this, use the steps to create a login from Unit 2 Lesson 4. Then, within the if statements of the while loop near return true add another if statement. This if statement should return true if the reader for the role column is equal to 'Teacher'. See the picture below for help.

*Remember to create a session variable for the username.

```csharp
while reader.Read()
    if Login1.UserName = reader(4) Then
        if Login1.Password = reader(5) Then
            if reader(3) = "Teacher" Then
                Session("login") = reader(4)
                return true
            End If
        End If
    End If
End If
```

5. Display the user's name at the top of the page using a label and the session variable that you created on the 'admin.aspx' page.

Label tag:

```xml
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
```

Sub Page Load (in the script tag):

```csharp
<script runat="server">
    Protected Sub Page_Load(sender As Object, ByVal e As System.EventArgs)
        Label1.Text = "Hi " & Session("Login")
    End Sub
</script>
```

6. Make a gridview that displays the 'schoollogin' table. Attach it to a sqldatasource and make sure to configure the datasource.

```xml
<asp:GridView ID="GridView1" runat="server" DataSourceID="SqlDataSource1">
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="% ConnectionStrings:connectionString%"
    ProviderName="% ConnectionStrings:connectionString.ProviderName%"
    SelectCommand="SELECT * FROM schoollogin"></asp:SqlDataSource>
```

7. Enable Paging and Sorting on your gridview.
8. Create a 5X3 table to organize the objects that you will need.
9. Drag in the following objects and set the properties as seen below. Double click each button to create a subroutine that runs when the button is clicked. The design and source code are provided below:

Design Mode:

```
Label

<table>
<thead>
<tr>
<th>Databound Col0</th>
<th>Databound Col1</th>
<th>Databound Col2</th>
</tr>
</thead>
<tbody>
<tr>
<td>abc</td>
<td>0</td>
<td>abc</td>
</tr>
<tr>
<td>abc</td>
<td>1</td>
<td>abc</td>
</tr>
<tr>
<td>abc</td>
<td>2</td>
<td>abc</td>
</tr>
<tr>
<td>abc</td>
<td>3</td>
<td>abc</td>
</tr>
<tr>
<td>abc</td>
<td>4</td>
<td>abc</td>
</tr>
</tbody>
</table>
```

Search by

Role: [Dropdown]

Id number: [Input]

First Name: [Input]

Last Name: [Input]

[Button: Search by Role]

[Button: Search by ID]

[Button: Search by First Name]

[Button: Search by Last Name]

Source Mode:

```xml
<table class="style1">
  <tr>
    <td class="style2">
      Search by</td>
    <td class="style3">
      &nbsp;</td>
    <td>
      &nbsp;</td>
  </tr>
  <tr>
    <td class="style2">
      Role: </td>
    <td class="style3">
      <asp:DropDownList ID="ddlrole" runat="server">
        <asp:ListItem>Both</asp:ListItem>
        <asp:ListItem>Teacher</asp:ListItem>
        <asp:ListItem>Student</asp:ListItem>
      </asp:DropDownList>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Button ID="btnRole" runat="server" Text="Search by Role" onclick="btnRole_Click" />
    </td>
  </tr>
</table>
```
10. Now, we need to limit the searches when the buttons are clicked. Let’s start with the ‘Search by Role’ button. First, go to the subroutine that is triggered when the button is clicked. Secondly, reset the select command by using this line of code:

```vbnet
Protected Sub btnRole_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * From schoollogin"
End Sub
```

This line of code gives us a blank slate to work off of each time the button is clicked.

11. The drop down list for the roles has three cases: Both, Teacher, or Student.
    - The first case, Both, has no effect on the original select command. Therefore, we do not need to write any code to account for this case.
• The second case, Teacher, does have an effect on the original select command. Add a where clause to the original select command that selects only Teachers from the database.
• The third case, Student, also has an effect on the original select command. Add a where clause that selects only Students from the database.

The code for the role button appears below: (Note that there are many other correct ways to do this)

```csharp
Protected Sub btnRole_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * From schoollogin"
    If ddlrole.Text = "Student" Then
        SqlDataSource1.SelectCommand += " where role='Student'"
    End If
    If ddlrole.Text = "Teacher" Then
        SqlDataSource1.SelectCommand += " where role='Teacher'"
    End If
End Sub
```

12. The final action that needs to be performed for the role button is to refresh the gridview. Therefore, add this line of code (GridViewID.DataBind()) at the end of the subroutine.

```csharp
Protected Sub btnRole_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * From schoollogin"
    If ddlrole.Text = "Student" Then
        SqlDataSource1.SelectCommand += " where role='Student'"
    End If
    If ddlrole.Text = "Teacher" Then
        SqlDataSource1.SelectCommand += " where role='Teacher'"
    End If
    GridView1.DataBind()
End Sub
```

13. Debug to make sure that this button works. Make sure to choose all 3 options from the dropdownlist.

14. Next, we will work on the ‘Search by Id’ button. We need to limit the search with a where statement that selects only the id’s between or equal to the two numbers of the textbox. Use concatenation, and your knowledge of sql to write a select command. Remember that because the id is a number, no tick marks are needed around the numbers in the select statement.

```csharp
Protected Sub btnID_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * from schoollogin where id= " & txtlowid.Text & " and id=" & & txthighid.Text
End Sub
```

16. Debug to test this button.

17. However, note that this code can easily crash if both textboxes are not filled in. Therefore, create a label next to the id button. This will be used as a warning label.

```csharp
<asp:Button ID="btnID" runat="server" onclick="btnID_Click">
    Text="Search by ID" />
<br />
<asp:Label ID="lblwarning" runat="server" Visible="false" text="Please enter numbers in both textboxes"></asp:Label>

The label should be invisible. Therefore, when we want the message to appear, you can set the label to visible.

18. Write an if, else statement that only allows these:

```csharp
Protected Sub btnID_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    If txtlowid.Text <> "" And txthighid.Text <> "" Then
        SqlDataSource1.SelectCommand = "Select * from schoollogin where id=" & txtlowid.Text & " and id<" & txthighid.Text
        GridView1.DataBind()
    Else
        lblwarning.Visible = True
    End If
End Sub
```

lines of code to run the low id and high id textboxes are not blank. Else, display the warning message.

19. Next, we will work on the first name button. The first name button is a simple concatenation as seen in the overview:

```csharp
Protected Sub btnfname_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * from schoollogin where firstname=" & txtfname.Text & ""
    GridView1.DataBind()
End Sub
```

*Remember to add GridViewID.DataBind()*

20. Next, we will work on the last name button. The last name button is similar to the first name button.

```csharp
Protected Sub btnlname_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "Select * from schoollogin where lastname=" & txtlname.Text & ""
    GridView1.DataBind()
End Sub
```

*Remember to add GridViewID.DataBind()*

21. Lastly, we will work on the reset button. First, reset the select command.

```csharp
Protected Sub btnReset_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "select * from schoollogin"
    GridView1.DataBind()
End Sub
```
22. Secondly, clear all of the textboxes (by setting the text inside the textboxes to "" or nothing)

```vbnet
Protected Sub btnReset_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.SelectCommand = "select * from schoollogin"
    GridView1.DataBind()

    'clear all of the textboxes
    txtfname.Text = ""
    txthighid.Text = ""
    txtlowid.Text = ""
    txtlname.Text = ""

    End Sub
```

**Individual Problems:**
Problem 2 is an individual problem. Please look at the Tips section and review the overview if you need help.

**Tips:**
- To search for a specific word within a record, read this page for more information:
  [http://www.w3schools.com/sql/sql_wildcards.asp](http://www.w3schools.com/sql/sql_wildcards.asp)
  - Use signs that allow you to search for anything on both sides of the word that you enter.
- Do not forget import namespace statements.
- Make sure that you have included all of your tick marks in the appropriate places. Remember that columns that are Integers do not need tick marks. However, if the data type of the column is VARCHAR then tick marks do need to be used.

**For More Information:**
Unit 2 Section 5 Pre-Test

Please take this test before the lesson.

Name:_________________________________

Directions: Please select a multiple choice option to answer the question.

1. ‘password’ is a column in the ‘student’ table of a database. Its data type is an integer. You want to execute one of the following select commands, using concatenation with a textbox called Textbox1. Which select command is correct (the sqldatasource is called sqldatasource1)?
   a. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text & “' ”
   b. Sqldatasource1.selectcommand= “Select * from student where password=’” & Textbox1.Text
   c. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text
   d. Sqldatasource1.selectcommand= “Select * from student where password=’” & Textbox1.Text & “’ ”

2. ‘password’ is a column in the ‘student’ table of a database. Its data type is VARCHAR. You want to execute one of the following select commands, using concatenation with a textbox called Textbox1. Which select command is correct (the sqldatasource is called sqldatasource1)?
   a. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text & “' ”
   b. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text
   c. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text
   d. Sqldatasource1.selectcommand= “Select * from student where password='” & Textbox1.Text & “’ ”

3. Which of the following statements will clear a textbox called Textbox1?
   a. Textbox1=""
   b. Textbox1.clear=true
   c. Textbox1.clear=false
   d. Textbox1.Text=""

4. What line of code resets a Gridview called Gridview1?
   a. Gridview1.DataReset()
   b. Gridview1.Reset()
   c. Gridview1.DataBind()
   d. Gridview1.BindData()
Unit 2 Section 5 Post-Test

Please take this test after the lesson. Answers

Name: ____________________________________

Directions: Please select a multiple choice option to answer the question.

1. ‘password’ is a column in the ‘student’ table of a database. Its data type is an integer. You want to execute one of the following select commands, using concatenation with a textbox called Textbox1. Which select command is correct (the sqldatasource is called sqldatasource1)?
   a. Sqldatasource1.selectcommand = "Select * from student where password='" & Textbox1.Text & "'
   b. Sqldatasource1.selectcommand = "Select * from student where password=" & Textbox1.Text
   c. Sqldatasource1.selectcommand = "Select * from student where password='" & Textbox1.Text
   d. Sqldatasource1.selectcommand = "Select * from student where password=" & Textbox1.Text & " '

2. ‘password’ is a column in the ‘student’ table of a database. Its data type is VARCHAR. You want to execute one of the following select commands, using concatenation with a textbox called Textbox1. Which select command is correct (the sqldatasource is called sqldatasource1)?
   a. Sqldatasource1.selectcommand = "Select * from student where password='" & Textbox1.Text & "'
   b. Sqldatasource1.selectcommand = "Select * from student where password=" & Textbox1.Text
   c. Sqldatasource1.selectcommand = "Select * from student where password='" & Textbox1.Text
   d. Sqldatasource1.selectcommand = "Select * from student where password=" & Textbox1.Text & " '

3. Which of the following statements will clear a textbox called Textbox1?
   a. Textbox1=""
   b. Textbox1.clear=true
   c. Textbox1.clear=false
   d. Textbox1.Text=""

4. What line of code resets a Gridview called Gridview1?
   a. Gridview1.DataBind()
   b. Gridview1.Reset()
   c. Gridview1.DataBind()
   d. Gridview1.DataBind()
Unit 2 Section 6: SQL Insert
Author: Hadiya
Date Created: 05-09-2014

Please take the Pre-Test before beginning this lesson.

Purpose:
The purpose of this lesson is to teach you how to insert records into the database. First, you will use insert statements within the database, then you will create a web application in which the user can add records to the database.

Overview:
Read over this link: http://www.w3schools.com/sql/sql_insert.asp

Try some insert statements in MySQL on your database. Do not move on to the problems until you can successfully insert a record into the database. Remember: many of the id’s in your tables are auto increment (AI). Therefore, you do not need to insert any id’s into your table. Also remember that your table should be referred to in the database as schema.table (ex. test.schoollogin).

Example:

```
*Note: a semicolon is required in the database, but is optional for sql commands in your websites.
*Also, click the first lightning button to perform the command that is written.

Problem Introduction:
Problem 1:
Display a gridview of the ‘schoollogin’ table. Remember to use a datasource! Use textboxes to insert data into the ‘firstname’, ‘lastname’, ‘username’, and ‘password’ fields. Use a dropdown list with the options (Student, Teacher) to insert into the ‘role’ field. Update the gridview after the user submits the data by clicking a submit button.

Problem 2:
Display a gridview of the ‘classes’ table. Remember to use a datasource! Use textboxes to insert data into the ‘classname’, ‘notes’, and ‘textbox’ fields. Update the gridview after the user submits the data by clicking a submit button.
Problem Programming Steps:

Problem 1:

1. Create a new website. We recommend that you call this website ‘sqlinsert’. Create a new page without code behind. We recommend that you call this page ‘schoollogin.aspx’.
2. Put your import namespace statements at the top of the page.
3. Use a gridview and sqldatasource to display the ‘schoollogin’ table. Enable paging and sorting.
4. Debug to make sure that your gridview is displaying data.
5. Create a 5X2 Table.
6. Use the following setup. The design and source code are provided.

Design:

![Table Design Image]

Insert Record:

First Name

Last Name

Role

Unbound

Username

Password

Submit

SqlDataSource - SqlDataSource1
Source Code:

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .style1
            { width: 43%; }
        .style2
            { width: 114px; }
    </style>
</head>
<body>
<form id="form1" runat="server">
    <div>
        <asp:GridView ID="GridView1" runat="server" AllowPaging="True" AllowSorting="True" DataSourceID="SqlDataSource1"></asp:GridView>
    </div>
    <br />
    <asp:SqlDataSource ID="SqlDataSource1" runat="server">
        <br />
        Insert Record:
        <table class="style1">
        <tr>
            <td class="style2">First Name</td>
            <td><asp:TextBox ID="txtfirstname" runat="server"></asp:TextBox></td>
        </tr>
        <tr>
            <td class="style2">Last Name</td>
            <td><asp:TextBox ID="txtlastname" runat="server"></asp:TextBox></td>
        </tr>
        <tr>
            <td class="style2">Role</td>
            <td><asp:DropDownList ID="ddlrole" runat="server"></asp:DropDownList></td>
        </tr>
        <tr>
            <td class="style2">Username</td>
            <td><asp:TextBox ID="txtusername" runat="server"></asp:TextBox></td>
        </tr>
    </table>
</form>
</body>
</html>
7. Setup the dropdownlist as described in the problem. The options should be student and teacher.

Design:

```
<asp:DropDownList ID="ddlrole" runat="server">
    <asp:ListItem>Teacher</asp:ListItem>
    <asp:ListItem>Student</asp:ListItem>
</asp:DropDownList>
```

8. Now, double click the button to create a subroutine that runs when the button is clicked. In this subroutine, we will place the code to create the insert statement and actually perform the insert. To create the insert statement, type the following statement that uses concatenation and +=.

```csharp
<asp:Button ID="btnSubmit" runat="server" onclick="btnSubmit_Click">
    Submit
</asp:Button>
```

```csharp
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.InsertCommand = "Insert into schoollogin (firstname, lastname, role, username, password) values ('" & txtfirstname.Text & ", " &
    SqlDataSource1.InsertCommand += txtlastname.Text & ", " &
    SqlDataSource1.InsertCommand += txtusername.Text & ", " &
    SqlDataSource1.InsertCommand += txtpassword.Text & ")"
End Sub
```
9. Response.Write the insert statement to make sure that there are no errors (tick marks, extra spaces etc.)

```csharp
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Response.Write(SqlDataSource1.InsertCommand)
    SqlDataSource1.Insert()
    GridView1.DataBind()
End Sub
```

10. Comment out or delete the Response.Write statement. Type the following code to actually perform the insert.

```csharp
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.Insert()
    GridView1.DataBind()
End Sub
```

11. Refresh the gridview so that the new record appears by typing the following code:

```csharp
SqlDataSource1.Insert()
GridView1.DataBind()
```

12. Debug.

Individual Problem:
Problem 2 is an individual problem. Please see the Tips section and the problem programming steps section if you need more help.

Tips:
- Do not forget import namespace statements.
- You do not have to write your insert statement all on one line. It is ok to write it on multiple lines using += or SqlDataSource1.insertCommand=sqlDataSource1.insert command & ...
- Make sure that you have included all of your tick marks in the appropriate places. Remember that columns that are Integers do not need tick marks. However, if the data type of the column is VARCHAR then tick marks do need to be used.
- There is no need to memorize the format of an insert command. An easy way to get the insert command format is to go to the database.
  1. Right click on the table you want an insert command for. Select copy to clipboard, then insert statement.
2. Open a new query tab by clicking this button:

3. Paste.

4. You can easily edit this into a usable SQL statement. First, take out all of the tick marks and delete the grey areas. The new information that you want to insert goes where the grey areas were. Don’t forget tick marks (if necessary)!!!
5. Also, the whole statement can be put all on one line for easy copying into visual web developer

```sql
INSERT INTO test.librarybooks
VALUES (12, 'Titanic', 'Ava Jones', 1995);
```

6. Or, you can try the statement in mysql by clicking the execute (lightning) button.

7. Check the bottom of the screen to see if the statement ran or errored.

For More Information:

Unit 2 Section 6 Pre-Test

Please take this test before the lesson.

Name: ____________________________

1. Which insert command is correct?
   b. INSERT into sports
   c. INSERT into sports (recordid, name, birthday, sport) values (1, ‘John Doe’, ‘July 8, 1954’, ‘Soccer’)
   d. INSERT into sports (recordid, name, birthday, sport values 1, ‘John Doe’, ‘July 8, 1954’, ‘Soccer’)

2. The columns ‘year’ and ‘productid’ are of the INTEGER datatype. ‘productid’ is a primary key. Which insert command is correct?
   a. INSERT into records (productid=2, year=2006)
   b. INSERT into records (productid= ‘2’, year=2006)
   c. INSERT into records (productid, year) values (2,’2006’)
   d. INSERT into records (productid, year) values (2, 2006)

3. The column ‘year’ is a VARCHAR datatype. The column ‘productid’ is an INTEGER data type and a primary key. Which insert command is correct?
   a. INSERT into records (productid=2, year=2006)
   b. INSERT into records (productid= ‘2’, year=2006)
   c. INSERT into records (productid, year) values (2,’2006’)
   d. INSERT into records (productid, year) values (2, 2006)

4. What does the final insert command for Sqldatasource1 look like?
   Sqldatasource1.InsertCommand= “Insert into records” & Textbox1.Text & Textbox2.Text
   Sqldatasource1.InsertCommand= “Insert into ”
   Sqldatasource1.InsertCommand+= “sports (name, sport) values ”
   Sqldatasource1.insertcommand+= “(’Doug’, ’Dodgeball’)”
   a. “Insert into records” & Textbox1.Text & Textbox2.Text
   b. “(’Doug’, ’Dodgeball’)”
   c. “Insert into records” & Textbox1.Text & Textbox2.Text & “Insert into sports (name, sport) values (’Doug’, ’Dodgeball’)”
   d. “into sports (name, sport) values ” & “(’Doug’, ’Dodgeball’)”
   e. “Insert into sports (name, sport) values (’Doug’, ’Dodgeball’)”

5. Is the insert command created by the lines of code in number 4 correct?
   a. Yes
   b. No
Unit 2 Section 6 Post-Test
Please take this test after the lesson. Answers

Name: ________________________________

1. Which insert command is correct?
   a. `INSERT into sports (recordid=1, name= 'John Doe', birthday= 'July 8, 1954', sport= 'Soccer')`
   b. `INSERT into sports`
   c. `INSERT into sports (recordid, name, birthday, sport) values (1, 'John Doe', 'July 8, 1954', 'Soccer')`
   d. `INSERT into sports (recordid, name, birthday, sport values 1, 'John Doe', 'July 8, 1954', 'Soccer')`

2. The columns ‘year’ and ‘productid’ are of the INTEGER datatype. ‘productid’ is a primary key. Which insert command is correct?
   a. `INSERT into records (productid=2, year=2006)`
   b. `INSERT into records (productid= '2', year=2006)`
   c. `INSERT into records (productid, year) values (2,'2006')`
   d. `INSERT into records (productid, year) values (2, 2006)`

3. The column ‘year’ is a VARCHAR datatype. The column ‘productid’ is an INTEGER data type and a primary key. Which insert command is correct?
   a. `INSERT into records (productid=2, year=2006)`
   b. `INSERT into records (productid= '2', year=2006)`
   c. `INSERT into records (productid, year) values (2,'2006')`
   d. `INSERT into records (productid, year) values (2, 2006)`

4. What does the final insert command for SqlDataSource1 look like?
   SqlDataSource1.InsertCommand= "Insert into records" & TextBox1.Text & TextBox2.Text
   SqlDataSource1.InsertCommand+= "Insert into "
   SqlDataSource1.insertcommand+= "(name, sport) values ("'
   a. "Insert into records" & TextBox1.Text & TextBox2.Text
   b. "'(Doug', 'Dodgeball')"
   c. "Insert into records" & TextBox1.Text & TextBox2.Text & "Insert into sports (name, sport) values ('Doug', 'Dodgeball')"
   d. "into sports (name, sport) values " & "'(Doug', 'Dodgeball')"
   e. "Insert into sports (name, sport) values ('Doug', 'Dodgeball')"

5. Is the insert command created by the lines of code in number 4 correct?
   a. Yes
   b. No
Unit 2 Section 7: SQL Select 5
Author: Hadiya
Date Created: 05-10-2014

Please take the Pre-Test before beginning this lesson.

Purpose: The purpose of this lesson is to teach you how to select data from a gridview so that the user can manipulate it using objects.

Overview:
Enable Selection (for a GridView)
Enabling selection puts a column on the side of a gridview that allows the user to select a row.

To enable selection, perform the following steps:

1. Click on your gridview. Then click the arrow next to it.
2. Click ‘Enable Selection’.

Extract the Selected Column from the Gridview

You can extract the data that a user selects from a gridview and use it in your application.

1. To find the code for this, go to documentation and search ‘gridview.selectedrow’.

The code example on this page appears like this:

The important code is in red.

```vbnet
Sub CustomersGridView_SelectedIndexChanged(ByVal sender As Object, ByVal e As EventArgs)

    ' Get the currently selected row using the SelectedRow property.
    Dim row As GridRowViewRow = CustomersGridView.SelectedRow

    ' Display the first name from the selected row.
    ' In this example, the third column (index 2) contains
    ' the first name.
    MessageLabel.Text = "You selected " & row.Cells(2).Text & ","

End Sub
```

However, we can condense this code down to this statement:

```vbnet
GridView1.SelectedValue.Cells(1).Text
```

- Replace `GridView1` with the name of your gridview
- Replace the number with the number of the column you would like to access. Column Numbering starts with the Select column as column 0.
Column Numbering:

<table>
<thead>
<tr>
<th>id</th>
<th>firstname</th>
<th>lastname</th>
<th>role</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 1</td>
<td>Perez</td>
<td>Love</td>
<td>Student</td>
<td>abcd123</td>
<td>Brain</td>
</tr>
<tr>
<td>Select 2</td>
<td>Numbers</td>
<td>Pages</td>
<td>Teacher</td>
<td>Microsoft</td>
<td>Word</td>
</tr>
<tr>
<td>Select 3</td>
<td>Mr.</td>
<td>Franks</td>
<td>Teacher</td>
<td>school</td>
<td>account</td>
</tr>
<tr>
<td>Select 6</td>
<td>Judy</td>
<td>Sun</td>
<td>Student</td>
<td>jsin</td>
<td>mynameisjudy</td>
</tr>
<tr>
<td>Select 7</td>
<td>life is hello</td>
<td>Teacher</td>
<td>iamateacher</td>
<td>password</td>
<td></td>
</tr>
<tr>
<td>Select 8</td>
<td>lax</td>
<td>Game</td>
<td>Student</td>
<td>softball</td>
<td>101</td>
</tr>
<tr>
<td>Select 10</td>
<td>a</td>
<td>b</td>
<td>Teacher</td>
<td>c</td>
<td>d</td>
</tr>
<tr>
<td>Select 11</td>
<td>Brian</td>
<td>Anderson</td>
<td>Teacher</td>
<td>BAnderson</td>
<td>ls4as7</td>
</tr>
</tbody>
</table>

*The Select column is column 0.
*id is column 1
*firstname is column 2
*lastname is column 3
*role is column 4
*username is column 5
*password is column 6

Therefore, to refer to the text that the user selected in a column, use the following statements:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Selected Row/Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GridView1.SelectedRow.Cells(1).Text</td>
</tr>
<tr>
<td>Firstname</td>
<td>GridView1.SelectedRow.Cells(2).Text</td>
</tr>
<tr>
<td>Lastname</td>
<td>GridView1.SelectedRow.Cells(3).Text</td>
</tr>
<tr>
<td>Role</td>
<td>GridView1.SelectedRow.Cells(4).Text</td>
</tr>
<tr>
<td>Username</td>
<td>GridView1.SelectedRow.Cells(5).Text</td>
</tr>
</tbody>
</table>

Example:
I can tell the user what they selected by using the statement:

Problem Introduction:

Problem 1:

Create a gridview that has selection, paging, and sorting enabled for the ‘schoollogin’ table. If the user selects a record, display all of the information related to the record that they created using Response.Write. See the overview for more help.

Problem 2:

Create a gridview that has selection, paging, and sorting enabled for the ‘classes’ table. If the user selects a record, display all of the information related to the record that they created using Response.Write. See the overview for more help.

Problem Programming Steps:

Problem 1:

1. Create a new website. We recommend that you call this website ‘gridviewselection’.
2. Create a new page without code behind. We recommend that you call this page ‘schoollogin.aspx’.
3. Create a gridview, connect it to a datasource, and configure the datasource.
4. Enable paging, sorting and selection.
5. Double click the gridview to create a Gridview1_SelectedIndexChanged Subroutine. This code will run every time the user selects a new record.

```vbnet
Protected Sub Gridview1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    End Sub
</script>
```

6. In the SelectedIndexChanged Subroutine type the code from the top of the page in the overview.

```vbnet
Protected Sub Gridview1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
End Sub
</script>
```

7. Debug. Click the select links to test your webpage.
Individual Problem:
Problem 2 is an individual problem. See the Tips and Overview sections if you need more help.

Tips:
- Remember to connect your gridview to the SQLDatasource.
- Remember to configure the SQLDatasource

For More Information:
Unit 2 Section 7 Pre-Test

Please take this test before the lesson.

Name: ________________________________

Directions: Use the following table to answer the questions. Answer each question by selecting a multiple choice option.

Select Statement: SELECT * FROM lamwrddl.librarian_table

<table>
<thead>
<tr>
<th>Librarian_ID</th>
<th>Librarian_Password</th>
<th>librarian_Last_Name</th>
<th>Librarian_First_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mylib2</td>
<td>midwest2</td>
<td>Stone</td>
<td>Sharon</td>
</tr>
<tr>
<td>mylib3</td>
<td>midwest3</td>
<td>Jones</td>
<td>Mary</td>
</tr>
<tr>
<td>mylib4</td>
<td>midwest4</td>
<td>Jackson</td>
<td>Samuel</td>
</tr>
<tr>
<td>mylib1</td>
<td>midwest1</td>
<td>Finnly</td>
<td>Sharon</td>
</tr>
</tbody>
</table>

1. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(1))
   What is printed out?
   a. Select
   b. mylib2
   c. mylib3
   d. midwest2

2. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(2))
   What is printed out?
   a. mylib2
   b. midwest2
   c. Stone
   d. mylib3

3. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(3))
   What is printed out?
   a. midwest2
   b. Stone
   c. Sharon
   d. mylib4

4. Which subroutine runs when the user clicks the Select links to the side of the Gridview?
   a. SQLDatasource1_SelectedIndexChanged
   b. GridView1_SelectedIndexChanged
   c. SQLDatasource1_SelectRow
   d. GridView1_SelectRow
Unit 2 Section 7 Post-Test

Please take this test after the lesson. [Answers]

Name: __________________________________________

Directions: Use the following table to answer the questions. Answer each question by selecting a multiple choice option.

Select Statement: SELECT * FROM lamwrdll.librarian_table

<table>
<thead>
<tr>
<th>Librarian_ID</th>
<th>Librarian_Password</th>
<th>librarian_Last_Name</th>
<th>Librarian_First_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mylib2</td>
<td>midwest2</td>
<td>Stone</td>
<td>Sharon</td>
</tr>
<tr>
<td>mylib3</td>
<td>midwest3</td>
<td>Jones</td>
<td>Mary</td>
</tr>
<tr>
<td>mylib4</td>
<td>midwest4</td>
<td>Jackson</td>
<td>Samuel</td>
</tr>
<tr>
<td>mylibr1</td>
<td>midwest1</td>
<td>Finny</td>
<td>Sharon</td>
</tr>
</tbody>
</table>

1. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(1))
   What is printed out?
   a. Select
   b. mylib2
   c. mylib3
   d. midwest2

2. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(2))
   What is printed out?
   a. mylib2
   b. midwest2
   c. Stone
   d. mylib3

3. The user selects the column where Librarian_ID= ‘mylib2’
   The following line of code runs:
   Response.Write(Gridview1.SelectedRow.Cells(3))
   What is printed out?
   a. midwest2
   b. Stone
   c. Sharon
   d. mylib4

4. Which subroutine runs when the user clicks the Select links to the side of the Gridview?
   a. SQLDatasource1_SelectedIndexChanged
   b. Gridview1_SelectedIndexChanged
   c. SQLDatasource1_SelectRow
   d. Gridview1_SelectRow
Unit 2 Section 8: SQL Delete  
Author: Hadiya  
Date Created: 05-10-2014

Please take the Pre-Test before starting this lesson.

Purpose:  
The purpose of this lesson is to teach you how to delete a record from a table using SQL.

Overview:  
Please read over the following page that explains the SQL Delete command:

http://www.w3schools.com/sql/sql_delete.asp

Next, try writing a delete command in MySql. An easy way to create a delete command in the database is below:

1. Go to MySQL Workbench.
2. Right click the table that you want to create a delete command for. Then click copy to Clipboard and Delete Statement.

3. Open a new sql tab.

4. Paste the Delete Statement (it has been placed on your clipboard already).

5. Delete the tick marks and the gray area near where.

DELETE FROM `test`.`schoollogin`  
WHERE `where_expression`;
6. Replace the gray area with a where expression. For example:

\[
\text{DELETE FROM test.schoollogin} \\
\text{WHERE id=11;}
\]

*Note: You usually want to delete using a where statement involving your primary key. This is because the primary key is unique; therefore your delete statement will only delete one record at a time.

7. Execute your Statement by clicking the lightning button.

8. If the statement worked correctly, you will see a positive result below. Also, you can select that table to check if the delete statement was correct.

*For delete commands do not forget the where clause. If you do, you could delete an entire table by accident.

**Problem Introduction:**

**Problem 1:**

Use your ‘schoollogin.aspx’ page from Unit 2 Section 7. When a user selects a record, make a Delete button appear. If the user clicks the button, delete the selected record (use a where clause with the primary key) and make the button invisible.

**Problem 2:**

Use the page you created for Problem 2 of Unit 2 Section 7. When a user selects a record, make a Delete button appear. If the user clicks the button, delete the selected record (use a where clause with the primary key) and make the button invisible.

**Problem Programming Steps:**

1. Open the page ‘schoollogin.aspx’ page in the website gridview selection.
2. Debug to make sure that the code still works.
3. Create a new button. The ID should be ‘btnDelete’. The text should be ‘Delete’. The property ‘Visible’ should be set to ‘False’. Double click the button to create a new subroutine called ‘btnDelete_Click’.

```html
<asp:Button ID="btnDelete" runat="server" Text="Delete" onclick="btnDelete_Click" Visible="false" />
```

4. Delete the Response.Write statements from the Gridview1.SelectedIndexChanged Subroutine.
Before:

```vbnet
Protected Sub Gridview1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    Response.Write("You selected id: " & Gridview1.SelectedRow.Cells(1).Text & "<br/>")
    Response.Write("You selected firstname: " & Gridview1.SelectedRow.Cells(2).Text & "<br/>")
    Response.Write("You selected lastname: " & Gridview1.SelectedRow.Cells(3).Text & "<br/>")
    Response.Write("You selected username: " & Gridview1.SelectedRow.Cells(5).Text & "<br/>")
End Sub
```

After:

```vbnet
Protected Sub Gridview1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
End Sub
```

5. In this subroutine, the code runs whenever the user selects a new record. Therefore, we should make the delete button visible in this subroutine.

```vbnet
Protected Sub Gridview1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    btnDelete.Visible = True
End Sub
```

6. Now, we will start on the Delete button. First, the delete button must collect the id of the record that the user selected.

```vbnet
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim id As Integer
    id = Convert.ToInt32(Gridview1.SelectedRow.Cells(1).Text)
End Sub
```

7. Secondly, the delete button must create an appropriate delete statement (for the SqlDataSource).

```vbnet
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim id As Integer
    id = Convert.ToInt32(Gridview1.SelectedRow.Cells(1).Text)
    'no tick mark needed because id is a number'
    SqlDataSource1.DeleteCommand = "Delete from schollongin where id=" & Convert.ToString(id)
End Sub
```

8. Thirdly, the delete button must perform the delete.
9. Fourthly, the delete button must refresh the data displayed in the gridview.

```vbnet
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim id As Integer
    id = Convert.ToInt32(Gridview1.CurrentRow.Cells(1).Text)
    'no tick mark needed because id is a number
    SqlDataSource1.DeleteCommand = "Delete from schoollogin where id=" & Convert.ToString(id)
    SqlDataSource1.Delete()
    Gridview1.DataBind()
End Sub
```

10. Lastly, the delete button must make itself invisible. If this does not happen, the user can cause the code to error by not selecting another record.

```vbnet
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim id As Integer
    id = Convert.ToInt32(Gridview1.CurrentRow.Cells(1).Text)
    'no tick mark needed because id is a number
    SqlDataSource1.DeleteCommand = "Delete from schoollogin where id=" & Convert.ToString(id)
    SqlDataSource1.Delete()
    Gridview1.DataBind()
    btnDelete.Visible = False
End Sub
```

11. Debug your application.

**Individual Problem:**
Problem 2 is an individual problem. See the Tips and Overview sections if you need help.
Tips:
- Do not forget the where clause in your delete commands.
- Make sure that you connect your gridview to your sqldatasource.
- Make sure that you configure your datasource.
  - There should be no brackets [] in your select statement.

For More Information:
- [http://www.w3schools.com/sql/sql_delete.asp](http://www.w3schools.com/sql/sql_delete.asp)
Unit 2 Section 8 Pre-Test

Please take this test before the lesson.

Name: __________________________

Directions: Please answer the following questions by selecting a multiple choice option.

1. Which SQL Delete Statement is correct?
   a. Delete * From studenttable
   b. Delete studentid=10436 From studenttable
   c. Delete From studenttable where studentid=10436
   d. Delete studenttable

2. Which type of column should be used in the where clause of a delete command?
   a. Friend key
   b. Primary key
   c. Binary
   d. Octal

3. Which delete command will delete a complete table (table name is ‘table’)?
   a. Delete from table
   b. Delete * from table
   c. Delete everything from table
   d. Delete table

4. Which delete command will delete the highlighted record?

   Table name: login

<table>
<thead>
<tr>
<th></th>
<th>id</th>
<th>firstname</th>
<th>lastname</th>
<th>role</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>1</td>
<td>Perez</td>
<td>Love</td>
<td>Student</td>
<td>abcde123</td>
<td>Brain</td>
</tr>
<tr>
<td>Select</td>
<td>2</td>
<td>Numbers</td>
<td>Pages</td>
<td>Teacher</td>
<td>Microsoft</td>
<td>Word</td>
</tr>
<tr>
<td>Select</td>
<td>3</td>
<td>Mr.</td>
<td>Franks</td>
<td>Teacher</td>
<td>school</td>
<td>account</td>
</tr>
<tr>
<td>Select</td>
<td>6</td>
<td>Judy</td>
<td>Sun</td>
<td>Student</td>
<td>jsun</td>
<td>mynameisjudy</td>
</tr>
<tr>
<td>Select</td>
<td>7</td>
<td>life is</td>
<td>hello</td>
<td>Teacher</td>
<td>iamateacher</td>
<td>password</td>
</tr>
<tr>
<td>Select</td>
<td>8</td>
<td>lax</td>
<td>Game</td>
<td>Student</td>
<td>softball</td>
<td>101</td>
</tr>
</tbody>
</table>

   a. Delete (2, Numbers, Pages, Teacher, Microsoft, Word)
   c. Delete from login where id=2
   d. Delete login where id=2
Unit 2 Section 8 Post-Test
Please take this test after the lesson. [Answers]

Name: ______________________________________

Directions: Please answer the following questions by selecting a multiple choice option.

1. Which SQL Delete Statement is correct?
   a. Delete * From studenttable
   b. Delete studentid=10436 From studenttable
   c. Delete From studenttable where studentid=10436
   d. Delete studenttable

2. Which type of column should be used in the where clause of a delete command?
   a. Friend key
   b. Primary key
   c. Binary
   d. Octal

3. Which delete command will delete a complete table (table name is 'table')?
   a. Delete from table
   b. Delete * from table
   c. Delete everything from table
   d. Delete table

4. Which delete command will delete the highlighted record?
   Table name: login

<table>
<thead>
<tr>
<th>id</th>
<th>firstname</th>
<th>lastname</th>
<th>role</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 1 Perez</td>
<td>Love</td>
<td>Student</td>
<td>abcd123</td>
<td>Brain</td>
<td></td>
</tr>
<tr>
<td>Select 2 Numbers</td>
<td>Pages</td>
<td>Teacher</td>
<td>Microsoft</td>
<td>Word</td>
<td></td>
</tr>
<tr>
<td>Select 3 Mr.</td>
<td>Franks</td>
<td>Teacher</td>
<td>school</td>
<td>account</td>
<td></td>
</tr>
<tr>
<td>Select 6 Judy</td>
<td>Sun</td>
<td>Student</td>
<td>jsun</td>
<td>mynameisjudy</td>
<td></td>
</tr>
<tr>
<td>Select 7 life is</td>
<td>hello</td>
<td>Teacher</td>
<td>iameatache</td>
<td>password</td>
<td></td>
</tr>
<tr>
<td>Select 8 lax</td>
<td>Game</td>
<td>Student</td>
<td>softball</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>

   a. Delete (2, Numbers, Pages, Teacher, Microsoft, Word)
   c. Delete from login where id=2
   d. Delete login where id=2
Unit 2 Section 9: SQL Update

Author: Hadiya
Date Created: 05-10-2014

Please take the Pre-Test before beginning this lesson.

Purpose:
The purpose of this lesson is to teach you how to update records using SQL.

Overview:
Please read the update lesson from WW3 Schools using the link below:

http://www.w3schools.com/sql/sql_update.asp

Next, try writing a update command in MySql. An easy way to create a delete command in the database is below:

10. Right click the table that you want to create an update command for. Then click copy to Clipboard and Update Statement.

11. Open a new SQL tab.

12. Paste the Update Statement (it has been placed on your clipboard already).
13. Delete the tick marks and the gray areas.

```sql
UPDATE test.schoollogin
SET
id = ,
firstname = ,
lastname = ,
role = ,
username = ,
password = ,
WHERE id = ;
```

14. Fill in data for where the grey areas were. Do not forget to do the where statement. If you do not include the where statement you could update the whole table.

```sql
UPDATE test.schoollogin
SET
id = 2,
firstname = 'Emma',
lastname = 'Loveland',
role = 'Teacher',
username = 'Elove',
password = '4356@bdyr'
WHERE id = 2 ;
```

*Note that the where statement usually uses the primary key of the table so that only one record is updated.

15. Execute your Statement by clicking the lightning button.

![Lightning bolt]

16. If the statement worked correctly, you will see a positive result at the bottom of the page. Also, you can select that table to check if the update statement was correct.

Problem Introduction:

**Problem 1:**

Create an update for all columns of the ‘schoollogin’ table except id. Use textboxes to update all columns except for the ‘role’ column. Use a drop down list to update the ‘role’ column. The update should be performed when the user clicks the update button.

**Problem 2:**
Create an update for all columns of the ‘librarybooks’ table except for the ‘bookid’ column. Use textboxes to update all of the columns. The update should be performed when the user clicks the update button.

Problem Programming Steps:

Problem 1:

1. Create a new website. We recommend that you name this website ‘sqlupdate’. Create a new webpage without code behind (all of your webpages should not have code behind). We recommend that you call this page ‘schoolloginupdate.aspx’.
2. Write import namespace statements at the top of the page.

```vb
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.Odbc" %>
```

3. Create a gridview and sqldatasource that displays the ‘schoollogin’ table. Enable paging, sorting, and selection for the gridview.

```xml
<asp:GridView ID="Gridview1" runat="server" AllowPaging="true"
    AllowSorting="true" DataSourceID="SqlDataSource1">
    <Columns>
        <asp:CommandField ShowSelectButton="true" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:connectionString %>">
    ProviderName="<%$ ConnectionStrings:connectionString.ProviderName %>">
    SelectCommand="SELECT * FROM schoollogin"</asp:SqlDataSource>
```

4. Create a webpage that looks like this: The source code and Design view are provided.

Design View (under the gridview):

![Design View Image]

The HTML code under the gridview and SQL Datasource:
5. Double click the gridview in design mode to create a Gridview1_SelectedIndexChanged Subroutine. In this subroutine, write code that will make the Update button visible.

6. Next, still inside the Gridview1_SelectedIndexChanged Subroutine, take the text from the gridview and insert it into the appropriate textboxes and/or dropdown lists.

*Make sure that you use the selected value property of the drop down list. The text property does not work for this example.

7. In the same subroutine, make the update button visible.
8. In the btnUpdate_Click Subroutine, create an update statement (for the SqlDataSource) using concatenation. Do not update the id column.

```vbnet
Protected Sub btnUpdate_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.UpdateCommand = "Update schoollogin set firstname='' & txtFirstname.Text"
    SqlDataSource1.UpdateCommand += ") lastnmae='' & txtLastname.Text"
    SqlDataSource1.UpdateCommand += ") role='' & ddlRole.Text"
    SqlDataSource1.UpdateCommand += ") username='' & txtUsername.Text"
    SqlDataSource1.UpdateCommand += ") password='' & txtPassword.Text & ''"
    SqlDataSource1.UpdateCommand += "where id='' & lblid.Text"
End Sub
```

9. Secondly, write code that actually performs the update.

```vbnet
Protected Sub btnUpdate_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.UpdateCommand = "Update schoollogin set firstname='' & txtFirstname.Text"
    SqlDataSource1.UpdateCommand += ") lastnmae='' & txtLastname.Text"
    SqlDataSource1.UpdateCommand += ") role='' & ddlRole.Text"
    SqlDataSource1.UpdateCommand += ") username='' & txtUsername.Text"
    SqlDataSource1.UpdateCommand += ") password='' & txtPassword.Text & ''"
    SqlDataSource1.UpdateCommand += "where id='' & lblid.Text"
    SqlDataSource1.Update()"
End Sub
```
10. Thirdly, refresh the gridview.

```vbnet
Protected Sub btnUpdate_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.UpdateCommand = "Update schoollogin set firstname='" & txtfirstname.Text
SqlDataSource1.UpdateCommand += "]', lastname='" & txtlastname.Text
SqlDataSource1.UpdateCommand += "]', role='" & ddlRole.Text
SqlDataSource1.UpdateCommand += "]', username='" & txtUsername.Text
SqlDataSource1.UpdateCommand += "]', password='" & txtPassword.Text & "]"
SqlDataSource1.UpdateCommand += "where id='" & lblid.Text

SqlDataSource1.Update()
GridView1.DataBind()
End Sub
```

11. Lastly, make the update button invisible (so that users cannot update without selecting a record).

```vbnet
Protected Sub btnUpdate_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SqlDataSource1.UpdateCommand = "Update schoollogin set firstname='" & txtfirstname.Text
SqlDataSource1.UpdateCommand += "]', lastname='" & txtlastname.Text
SqlDataSource1.UpdateCommand += "]', role='" & ddlRole.Text
SqlDataSource1.UpdateCommand += "]', username='" & txtUsername.Text
SqlDataSource1.UpdateCommand += "]', password='" & txtPassword.Text & "]"
SqlDataSource1.UpdateCommand += "where id='" & lblid.Text

SqlDataSource1.Update()
GridView1.DataBind()
btnUpdate.Visible=False
End Sub
```

12. Debug.

Individual Problem:
Problem 2 is an individual problem. See the Overview, Problem Programming Steps, and Tips sections if you need help.

Tips:
- Do not forget the where clause in your update statement.
- Remember to write import namespace statements.

For More Information:
- [http://www.w3schools.com/sql/sql_update.asp](http://www.w3schools.com/sql/sql_update.asp)
Unit 2 Section 9 Pre-Test

Please take this test before the lesson.

Name: ______________________________________

Directions: Please select a multiple choice answer to the question.

1. Which update statement is correct?
   a. UPDATE into table (role= ‘Admin’, username= ‘admin’, password= ‘admin’)
   b. UPDATE * into table where id=2 (role= ‘Admin’, username= ‘admin’, password= ‘admin’)
   c. UPDATE table (role= ‘Admin’, username= ‘admin’, password= ‘admin’) where id=2
   d. UPDATE table set role= ‘Admin’, username= ‘admin’, password= ‘admin’ where id=2

2. Which button executes a SQL statement in MySQL Workbench?
   a. 
   b. 
   c. 
   d. 

3. Which update statement updates the entire table?
   a. UPDATE table set role= ‘Admin’, username= ‘admin’, password= ‘admin’
   b. UPDATE * into table where id=2 role= ‘Admin’, username= ‘admin’, password= ‘admin’
   c. UPDATE table (role= ‘Admin’, username= ‘admin’, password= ‘admin’) where id=2
   d. UPDATE table set role= ‘Admin’, username= ‘admin’, password= ‘admin’ where id=2

4. What field creates a new SQL file in MySQL Workbench?
   a. 
   b. 
   c. 
   d. 
Unit 2 Section 9 Post-Test
Please take this test after the lesson. Answers

Name: ____________________________________________

Directions: Please select a multiple choice answer to the question.

1. Which update statement is correct?
   a. UPDATE into table (role= 'Admin', username= 'admin', password= 'admin')
   b. UPDATE * into table where id=2 (role= 'Admin', username= 'admin', password= 'admin')
   c. UPDATE table (role= 'Admin', username= 'admin', password= 'admin') where id=2
   d. UPDATE table set role= 'Admin', username= 'admin', password= 'admin' where id=2

2. Which button executes a SQL statement in MySQL Workbench?
   a. 
   b. 
   c. 
   d. 

3. Which update statement updates the entire table?
   a. UPDATE table set role= 'Admin', username= 'admin', password= 'admin'
   b. UPDATE * into table where id=2 role= 'Admin', username= 'admin', password= 'admin'
   c. UPDATE table (role= 'Admin', username= 'admin', password= 'admin') where id=2
   d. UPDATE table set role= 'Admin', username= 'admin', password= 'admin' where id=2

4. What field creates a new SQL file in MySQL Workbench?
   a. 
   b. 
   c. 
   d. 

Unit 2 Section 10: SQL Combination (Unit 2 Review)
Author: Hadiya
Date Created: 05-11-2014

There is no Pre-Test or Post-Test for this section because it is a review section.

Purpose:
The purpose of this lesson is for you to review Unit 2. You will combine all of the knowledge you have gained from this Unit (SQL Select, SQL Insert, SQL Update, SQL Delete, and Logins).

Overview:
Below you will find a code walkthrough of a webpage that allows users to insert, update, and delete records from the schoollogin table.

Design View:
Source Code/Code Walkthrough:

Note that you may want to review the HTML Section of the Code before reading the Script Section.

<table>
<thead>
<tr>
<th>Lines of Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;%@ Page Language=&quot;VB&quot; %&gt;</code></td>
<td>Import Namespace Statements</td>
</tr>
<tr>
<td><code>&lt;%@ Import Namespace=&quot;System.Data&quot; %&gt;</code></td>
<td>Import Namespace Statements</td>
</tr>
<tr>
<td><code>&lt;%@ Import Namespace=&quot;System.Data.Odbc&quot; %&gt;</code></td>
<td>Import Namespace Statements</td>
</tr>
<tr>
<td>`&lt;!DOCTYPE html PUBLIC &quot;-//W3C//DTD XHTML 1.0 Transitional//EN&quot;</td>
<td>Starting Tag for Script</td>
</tr>
<tr>
<td>&quot;<a href="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd%22%3E">http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd&quot;&gt;</a></td>
<td>Gridview1_SelectedIndexChanged</td>
</tr>
<tr>
<td><code>&lt;script runat=&quot;server&quot;&gt;</code></td>
<td>When one of the select buttons is clicked, the ID number of the selected record is printed out.</td>
</tr>
<tr>
<td>```vbnet<code>Protected Sub GridView1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)</code></td>
<td>Then the Delete and Update buttons are made available/visible for the user to see because they have selected a record.</td>
</tr>
<tr>
<td>Response.Write(&quot;You have selected id number: &quot; &amp; GridView1.SelectedRow.Cells(1).Text &amp; &quot;&lt;br/&gt;&quot;) `</td>
<td>If the user wants to delete, then the user will click the delete button.</td>
</tr>
<tr>
<td>'make the next step label and buttons visible to instruct the user on what to do next `</td>
<td>Concatenation to create the delete command.</td>
</tr>
<tr>
<td>lblQuestion.Visible = True `</td>
<td>Id is column 1 in the gridview. No tick marks are needed because id is a number.</td>
</tr>
<tr>
<td>btnDelete.Visible = True `</td>
<td>Then the delete is performed.</td>
</tr>
<tr>
<td>btnUpdate.Visible = True `</td>
<td>Make the options to delete and update invisible (until the user selects another record. This is controlled by the Gridview1_SelectedIndexChanged Subroutine)</td>
</tr>
<tr>
<td>End Sub `</td>
<td></td>
</tr>
</tbody>
</table>
If the user wants to update, the user will click the update button.

This section makes all of the textboxes, labels, buttons, and drop down lists associated with the update visible.

If the user wants to cancel the update, they can click the Cancel button.

This section makes the other buttons invisible.

If the user wants to finish the update, they can click the Submit button.

This section puts the values in the gridview into the textboxes and dropdown list.

If the user clicks the Submit button, then this code will run.
to get sql command, right click on schoollogin
table, then fill in the appropriate data
' use concatenation to get data from the gridview
' no need to do id because it is set to AI
' the sql command from mysql has tickmarks in
random places
' need tickmarks around text
SqlDataSource1.UpdateCommand = "Update
test.schoollogin Set firstname='" & txtfname.Text & ", " &
txtlname.Text & ", " &
SqlDataSource1.UpdateCommand += "lastname='" &
ddlrole.Text & ", " &
SqlDataSource1.UpdateCommand += "role='" &
ddlusername.Text & ", " &
SqlDataSource1.UpdateCommand += "username='" &
txtusername.Text & ", " &
SqlDataSource1.UpdateCommand += "password='" &
txtpassword.Text & ", "
' id is a number, therefore no tick marks
SqlDataSource1.UpdateCommand += "where id=" &
GridView1.SelectedRow.Cells(1).Text
SqlDataSource1.Update()'
'reflect the changes in the gridview
GridView1.DataBind()
'remove the textboxes/dropdownlists/labels
lblfname.Visible = False
txtfname.Visible = False
lbllname.Visible = False
txtlname.Visible = False
lblrole.Visible = False
ddlrole.Visible = False
lblusername.Visible = False
txtusername.Visible = False
lblpassword.Visible = False
txtpassword.Visible = False
btnSubmit.Visible = False
lblid.Visible = False
btnCancel.Visible = False
End Sub

Protected Sub btnCancel_Click(ByVal sender As Object,
ByVal e As System.EventArgs)
' remove the textboxes/dropdownlists/labels
lblfname.Visible = False
txtfname.Visible = False
lbllname.Visible = False
txtlname.Visible = False
lblrole.Visible = False
ddlrole.Visible = False
lblusername.Visible = False
txtusername.Visible = False
lblpassword.Visible = False
txtpassword.Visible = False
btnSubmit.Visible = False
lblid.Visible = False
btnCancel.Visible = False
End Sub

The update command is created here using concatenation.

Don't forget the where clause.

Refresh the data in the gridview after updating.

Make all of the buttons, labels, textboxes, and dropdown lists associated with the update invisible.

If the user clicks the Cancel button, then this code will run.

Make all of the buttons, labels, textboxes, and dropdown lists associated with the update invisible.
If the user wants to insert a record, they will click the insert button.

Create the insert command using concatenation. Use tick marks for the columns that have text.

Notice that the textboxes and dropdown list in the insert have a zero at the end to distinguish from the textboxes and dropdown list used in the update.

Perform the insert.

Refresh the gridview.

Delete the text from the textboxes.

End the Script section

Begin the HTML Section

Start the body.

This first div is for the delete and update.

Instruct the user to select a record

Note that \&nbsp; means space.

The Gridview that displays the schoollogin table

Note that Paging and Sorting are enabled

Remember to connect the gridview to a SQL DataSource
When the user selects something, the subroutine **GridView1_SelectedIndexChanged** runs (see the script section).

Note that Selection is enabled.

This label appears whenever the Update and Delete buttons are visible. The label instructs the user to click one of the buttons.

The label is invisible when the page first loads.

The Update button appears after the user selects a new option in the gridview.

The button is invisible when the page first loads.

When the button is clicked the **btnUpdate_Click** Subroutine runs.

The Delete button appears after the user selects a new option in the gridview.

The button is invisible when the page first loads.

When the button is clicked the **btnDelete_Click** Subroutine runs.

These labels, textboxes, buttons, and drop down lists are for the update command.

They are initially invisible when the page first loads.
Note that the DropdownList has the items Teacher and Student added.

For the lblid label, the text of the id that is being updated is concatenated into the label in the script (see btnUpdate_Click in between the script tags).

<table>
<thead>
<tr>
<th>Note that the DropdownList has the items Teacher and Student added.</th>
</tr>
</thead>
<tbody>
<tr>
<td>For the lblid label, the text of the id that is being updated is concatenated into the label in the script (see btnUpdate_Click in between the script tags).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The submit button appears if the clicks the update button. When the user clicks submit, the record is updated.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The submit button is initially invisible when the page first loads. The subroutine btnSubmit_Click runs when the user clicks the submit button.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The cancel button btnCancel_Click runs when the user wants to cancel the update.</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the SQLDatasource that Gridview1 is connected to. Remember to configure the datasource.</td>
</tr>
</tbody>
</table>

```c
<asp:Label ID="lblname" runat="server" Text="Last Name: " Visible="false"></asp:Label>
<asp:TextBox ID="txtnname" runat="server" Visible="false"></asp:TextBox>
<br />
<asp:Label ID="lblrole" runat="server" Text="Role: " Visible="false"></asp:Label>
<asp:DropDownList ID="ddlrole" runat="server" Visible="false">  
  <asp:ListItem>Teacher</asp:ListItem>  
  <asp:ListItem>Student</asp:ListItem>  
</asp:DropDownList>
<br />
<asp:Label ID="lblusername" runat="server" Text="Username: " Visible="false"></asp:Label>
<asp:TextBox ID="txtusername" runat="server" Visible="false"></asp:TextBox>
<br />
<asp:Label ID="lblpassword" runat="server" Text="Password: " Visible="false"></asp:Label>
<asp:TextBox ID="txtpassword" runat="server" Visible="false"></asp:TextBox>
<br />
<asp:Button ID="btnSubmit" runat="server" Text="Submit" visible="false" onclick="btnSubmit_Click"/>
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />

Note that the DropdownList has the items Teacher and Student added.

For the lblid label, the text of the id that is being updated is concatenated into the label in the script (see btnUpdate_Click in between the script tags).

The submit button appears if the clicks the update button. When the user clicks submit, the record is updated.

The submit button is initially invisible when the page first loads. The subroutine btnSubmit_Click runs when the user clicks the submit button.

The cancel button btnCancel_Click runs when the user wants to cancel the update.

This is the SQLDatasource that Gridview1 is connected to. Remember to configure the datasource.
The second div is for the insert. I dragged the div to the right side of the screen by selecting the div, going to the format menu, selecting position, and setting the position to absolute. This allows you to drag the div around freely in design view.

If you would like to insert a new record, please fill out the information below:

These labels, textboxes, dropdown lists and buttons are associated with the insert statement.
The textboxes, labels, and dropdown lists have the same ID as their counterparts used for the update. However, there is an extra 0 on the end to distinguish them.

The submit button creates and performs the insert statement in a subroutine called bntSubmitinsert_Click.

This tag ends the div for the insert.
The ending HTML and body tags.
Problem Introduction:

Problem 1:
Create an administrative website for a school. Allow only teachers to login using the ‘schoollogin’ table.

After the teacher has logged in, greet them by username (using a session variable). Also, allow teachers to update, delete, and insert new records into the ‘schoollogin’ table. An example is provided in the overview.

Problem 2:
Create a website for a library. Use the ‘security’ table to create a login.

When the user logs in, greet them by firstname and lastname (using session variables). Allow the user to search for books in the ‘librarybooks’ table. Users should be able to search by the title of the book, author, and year published. The results should be displayed in a gridview.

Individual Problems:
Problems 1 and 2 are both individual problems. See the Overview for a code walkthrough of a problem similar to problem 1. Problem 2 is similar to both problems presented in Unit 2 Section 5. Also, review the Tips section for help.

Tips:
- Do not forget import namespace statements.
- Remember to connect gridviews to SQLDatasources.
- Configure each SQL Datasource. Remember that there should be no brackets in the select command.
- Remember to use concatenation for complex select, update, insert, and delete statements.

For More Information:
- http://www.w3schools.com/sql/

There is no Pre-Test or Post-Test for this lesson because it is a Review section.
Unit 3 Section 1: Arduino Stoplight Simulator

Date: 02/15/14
Authors: Hadiya, Brian, Kiese, Hasani & Roz

**Project:** Stop Light Simulator

Please complete the [Pre-Test](#) before starting.

**Purpose:** The purpose of this lesson is to demonstrate the capabilities of the Arduino microcontroller.

**Overview:** Each side of the Stop Light Simulator features three lights: red, green and yellow. The lights flash in a coordinated manner so that traffic is able to pass safely. Simulator features red, yellow and green flashing lights built into an electrical outlet box and connected to an Arduino microcontroller board.

The lights are connected to color coded wires: red light connected to the red wire, yellow light connected to gold wire, and green light connected to the green wire. There is also a black wire connected to ground which completes the circuit, so that electricity can flow. On one end of the lights there are resistors and color coded wires. On the other end the wires are connected to ground.

The Arduino Board circuitry includes pins. The role of the pins is to connect circuits that serve as input or output. The color coded wires are connected to the pins. The pins are numbered; therefore each light is associated with a number.

The left photo depicts the top of the board and the right photo depicts the bottom of the board.

Four of these boards are mounted inside of the stop light; one on each side.
This is the front of the stop light simulator. Left and right designations are based on this view.
Blue Simulator Pin Map

- **Left Lights**
  - Pin 2 - Red
  - Pin 3 - Yellow
  - Pin 4 - Green

- **Front Lights**
  - Pin 5 - Red
  - Pin 6 - Yellow
  - Pin 7 - Green

- **Right Lights**
  - Pin 8 - Red
  - Pin 9 - Yellow
  - Pin 10 - Green

- **Back Lights**
  - Pin 11 - Red
  - Pin 12 - Yellow
  - Pin 13 - Green
Programming Steps

1. Plug in Arduino into USB port.
2. Open the blink sketch by going to “File”, “examples”, “Basics”, “Blink”:

```cpp
void setup() {
  pinMode(red_front, HIGH); //initialize red_front pin as high
  delay(1000); //wait one second
  digitalWrite(red_front, LOW); //make red_front pin low
  delay(1000); //wait one second
}
```

3. The first few lines that are greyed out with “/*” and “//” are considered comments. These comments are not code and will not run.

```cpp
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/
```

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
4. The first orange line that says “int” is a variable. A variable is a word that assumes a value. Similar to “x” in algebra. In this case the variable is an integer. Therefore the line starts with “int”. The next word is “led” which is the name of the variable. Next you’ll see that “led=13”. This assigns led a value. Lastly there is a semicolon. This ends each line of code and is necessary and important.

\[
\text{int \ led = 13;}
\]

5. Next line in orange is “void setup()”. This is considered a function. This function will only run once. Next you’ll notice the brackets “{”. You need one set of these “{ }”. Everything in these brackets is a part of the function “setup()”.

```c
void setup() {
  // initialize the digital pin as output.
  pinMode(led, OUTPUT);
}
```

6. The “pinMode” function sets a requested pin as input or output mode. For this problem all the pins are connected to lights therefore all pins should be set to output. The format of this function is (pin number, mode). You will see that “led” is in the pin number 13. The mode is “OUTPUT”. This line of code will set pin number 13 as OUTPUT.

```c
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

7. Next is “void Loop()”. This function continuously runs after the “setup()” function. It is also needs brackets.

```c
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

8. Next is “digitalWrite”. This is used to turn the lights on and off. The format of this function is (pin number, value). The value can only be “HIGH” or “LOW”. “HIGH” turns the led on while “LOW” turns it off.

```c
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
9. Next is “delay”. This function is used to pause the program for a number of milliseconds. 1000 milliseconds equals 1 second. So, in the code here the program is waiting for one second until it goes on again.

```c
void loop() {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

Downloading the Program onto your Arduino Board:

1. To download the program onto your Arduino board, make sure that the USB cable is plugged into your board and computer.

2. Next, on the computer, go to the “Tools” menu. Then click “Board” and select your board type. In this case, we used the Arduino Duemilanove Board.

2. Next go to the “Tools” menu again. Then click “Serial Port” and select the correct serial port. If you do not know the correct serial port, choose “COM 4” and continue to the next steps. If you get errors that are unrelated to errors in your code, then try a different com port. In our example, we used COM 4.
4. Lastly, click the arrow at the top left corner of the screen to “Upload” your code to the Arduino.

At this point, the code will compile. When code compiles it is changed from the code that you typed in, into machine language. If any errors present themselves, double check your code for semicolons, double check your code for syntax errors, make sure your com port is correct (see step 12), and make sure that your board is correct (see step 11).

After the code compiles, it will be placed on your Arduino device and will run.

Tips: Writing comments will help make the code you write clear to you and others. Naming your pins wisely, for example: red_front, green_right, or yellow_back will also make your code clear. Make sure to put a semicolon after every function.

Problems:
Problem 1: Name all the lights by declaring them as integers. For example:

```cpp
int red_front = 2;
int yellow_front = 3;
int green_front = 4;
```

You have 12 lights total on your stoplight. Each needs to be declared as an integer that is equal to its corresponding pin number. See pin diagram for help. And then set them all as output in the function “setup()”. For help see step number 6.
Secondly, pick one of the led pins and make it blink by putting its corresponding name in place of “led”. For help see step number 8.

**Problem 2:** Create a program that will blink all of the red lights on and off at the same time.

**Problem 3:** Create a program that will blink all of the lights in order by pin number, one by one.

**Problem 4:** Create a maintenance program that will turn all lights on and off for one second.

**Problem 5:** Light up one side like a stop light

<table>
<thead>
<tr>
<th>Time (milliseconds)</th>
<th>Side 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Red</td>
</tr>
<tr>
<td>2000</td>
<td>Red</td>
</tr>
<tr>
<td>3000</td>
<td>Green</td>
</tr>
<tr>
<td>4000</td>
<td>Green + Yellow</td>
</tr>
</tbody>
</table>

For this problem you need to use the delay function. For help refer to step number 9.

**Problem 6:** Now light up the two opposite sides of the stoplight. Remember that the two opposite sides of the stoplight are exactly the same.

**Problem 7:** Make the stoplight more realistic. Incorporate the other two sides and use the grid below for timing.

<table>
<thead>
<tr>
<th>Time(milliseconds)</th>
<th>Side 1+Side 3</th>
<th>Side 2+Side 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>2000</td>
<td>Red</td>
<td>Green + Yellow</td>
</tr>
<tr>
<td>3000</td>
<td>Green</td>
<td>Red</td>
</tr>
<tr>
<td>4000</td>
<td>Green + Yellow</td>
<td>Red</td>
</tr>
</tbody>
</table>
For more information:

- [http://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination.html](http://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination.html)
- [http://vimeo.com/18539129](http://vimeo.com/18539129)
- [http://www.arduino.cc/](http://www.arduino.cc/)
Unit 3 Section 1 Arduino Stop Light Simulator Pre-Test

Please take this test before the lesson.

Name:_________________________________

Directions: Please explain what each line of code does by answering a multiple choice question for numbers 1-4.

1. Int front_red=13;
   a. Turn the front red light on
   b. Create an integer called front_red and assign it the value 13
   c. Integrate the number 13
   d. Interpret front_red

2. Pin 13 will be used as a light. What should be entered into the blank in the following line of code?
   void setup(){
       pinMode(13, _________);
   }
   a. OUTPUT
   b. ON
   c. LIGHT
   d. INPUT

3. Which line of code will make the program wait 1 second?
   a. delay(1000);
   b. delay(1000)
   c. delay(4000);
   d. delay(10000);

4. Pin 13 will be used as a light. The light should be turned on. What should be entered into the blank in the following line of code?
   void loop() {
       digitalWrite(13, ________);
   }
   a. LOW
   b. LIGHT
   c. ON
   d. HIGH

5. Please write your answer to the following question: What does the following line of code do?
   // turn on the light connected to pin 13
Unit 3 Section 1 Stop Light Simulator Post-Test

Please take this test after the lesson. Answers

Name: _____________________________________________

Directions: Please explain what each line of code does by answering a multiple choice question for numbers 1-4.

1. Int front_red=13;
   a. Turn the front red light on
   b. Create an integer called front_red and assign it the value 13
   c. Interpret front_red
   d. Integrate the number 13

2. Pin 13 will be used as a light. What should be entered into the blank in the following line of code?
   void setup(){
       pinMode(13, _________);
   }
   a. OUTPUT
   b. INPUT
   c. LIGHT
   d. ON

3. Which line of code will make the program wait 1 second?
   a. delay(1000)
   b. delay(1000);
   c. delay(4000);
   d. delay(10000);

4. Pin 13 will be used as a light. The light should be turned on. What should be entered into the blank in the following line of code?
   void loop() {
       digitalWrite(13, ________);
   }
   a. LOW
   b. LIGHT
   c. ON
   d. HIGH

5. Please write your answer to the following question: What does the following line of code do?
   // turn on the light connected to pin 13
Answers to Pre-Tests and Post-Tests

Unit 1 Section 1
- Pre-Test
  - 1 b
  - 2 c
  - 3 a
  - 4 d
  - 5 a
- Post-Test
  - 1 b
  - 2 c
  - 3 a
  - 4 d
  - 5 a

Unit 1 Section 2
- Pre-Test
  - 1 a
  - 2 c
  - 1 b
- Post-Test
  - 2 e
  - 3 c
  - 1 d

Unit 1 Section 3
- Pre-Test
  - 1 d
  - 2 b
  - 3 e
- Post-Test
  - 1 d
  - 2 b
  - 3 e

Unit 1 Section 4
- Pre-Test
  - 1 d
  - 2 b
  - 3 a
- Post-Test
  - 4 d
  - 5 b
  - 6 a

Unit 1 Section 5
- Pre-Test
  - 12-Integer(Int32)
  - Hello-String
  - True-Boolean
  - 12.2-Decimal
  - 1 d
  - 2 b
  - 3 a
- Post-Test
  - 12-Integer(Int32)
  - Hello-String
  - True-Boolean
  - 12.2-Decimal
  - 1 d
  - 2 b
  - 3 a

Unit 1 Section 6
- Pre-Test
  - 1 a
  - 2 d
  - 3 c
  - 4 d
  - 5 a
- Post-Test
  - 1 a
  - 2 d
  - 3 c
  - 4 d
  - 5 a

Unit 1 Section 7
- Pre-Test
  - 1 d
  - 2 a
  - 3 b
  - 4 c
  - 5 d
  - 6 a
- Post-Test
  - 1 d
  - 2 a
<table>
<thead>
<tr>
<th>Unit 1 Section 8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 b</td>
</tr>
<tr>
<td></td>
<td>4 c</td>
</tr>
<tr>
<td></td>
<td>5 d</td>
</tr>
<tr>
<td></td>
<td>6 a</td>
</tr>
<tr>
<td>➢ Pre-Test</td>
<td>1 d</td>
</tr>
<tr>
<td></td>
<td>2 b</td>
</tr>
<tr>
<td></td>
<td>3 d</td>
</tr>
<tr>
<td></td>
<td>4 b</td>
</tr>
<tr>
<td></td>
<td>5 a</td>
</tr>
<tr>
<td>➢ Post-Test</td>
<td>1 d</td>
</tr>
<tr>
<td></td>
<td>2 b</td>
</tr>
<tr>
<td></td>
<td>3 d</td>
</tr>
<tr>
<td></td>
<td>4 a</td>
</tr>
<tr>
<td></td>
<td>5 a</td>
</tr>
<tr>
<td>➢ Post-Test</td>
<td>1 d</td>
</tr>
<tr>
<td></td>
<td>2 b</td>
</tr>
<tr>
<td></td>
<td>3 d</td>
</tr>
<tr>
<td></td>
<td>4 a</td>
</tr>
<tr>
<td></td>
<td>5 a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 1 Section 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Pre-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>➢ Post-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 1 Section 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Pre-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>➢ Post-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 2 Section 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Pre-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>➢ Post-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 2 Section 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Pre-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>➢ Post-Test</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 2 Section 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Pre-Test</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Unit 2 Section 5

- Pre-Test
  - 1 a
  - 2 a
  - 3 a
  - 4 a
- Post-Test
  - 1 a
  - 2 a
  - 3 a
  - 4 a

Unit 2 Section 6

- Pre-Test
  - 1 b
  - 2 a
  - 3 c
  - 4 c
- Post-Test
  - 1 b
  - 2 a
  - 3 d
  - 4 c

Unit 2 Section 7

- Pre-Test
  - 1 d
  - 2 c
  - 3 c
  - 4 b
- Post-Test
  - 1 d
  - 2 c
  - 3 c

Unit 2 Section 8

- Pre-Test
  - 1 a
  - 2 a
  - 3 a
  - 4 c
- Post-Test
  - 1 a
  - 2 a
  - 3 a

Unit 2 Section 9

- Pre-Test
  - 1 c
  - 2 b
  - 3 a
  - 4 e
- Post-Test
  - 1 d
  - 2 c
  - 3 a

Unit 2 Section 10 (Unit 2 Review)

- No Pre-Test or Post-Test because this is a review lesson

Unit 3 Section 1

- Pre-Test
  - 1 b
  - 2 a
  - 3 a
  - 4 d
  - 5. This line is a comment, therefore it does nothing.
- Post-Test
  - 1 b
  - 2 a
  - 3 b
  - 4 d
  - 5. This line is a comment, therefore it does nothing.