IST 256
Lab Week 2 – September 9, 2009

Your name: ____

1. Thinking about Constants and Variables

a. For each constant value in the list below, give the most likely Visual Basic type for that value, using the three types: Integer, Single, and String

219 ____________________
16.54 ____________________
“To be or not to be” ____________________
- 32.0 ____________________
- 3 ____________________
“Otto the Orange” ____________________

b. Suppose that we have declared a variable named “saleitems” as shown:

```
Dim saleitems as Integer
```

Then we have an assignment statement:

```
saleitems = 1
```

What is the value of the variable in memory after this assignment? ________________

Next we have another assignment statement:

```
saleitems = saleitems + 2
```

What is the value of the variable in memory after this assignment? _________________

2. Writing a program with a counter.

1. Start a new project: under the project menu, start a new project and call it something like “Counting”.

2. Add one Button to the Form. Change the Text property of the Form (e.g. “Count Program”) and also the Text property of the Button (e.g. “Count Numbers”) and make whatever other changes to Fonts and Colors of the Button that you like.

3. Write the Button Program. Double click on the Button on the Form to get the Form1.vb program window to appear.

First, after the top line that starts with Public Class … , add lines that say

```
Dim count as Integer = 0
Dim message as String
```

Count is a variable that will hold numbers to count and will start with the value 0. Message is a variable that will hold a string with a message and the value of count.

Now in-between the lines that start with Sub and End Sub, add the following lines. You can either type them in, or you can go to the on-line instructions in the iLMS system and copy and paste these lines.

```
count = count + 1
message = “You have clicked the button “ & count & “ times”
MsgBox(message)
```

This will add 1 to the count number each time the button is clicked and pop up the message.

4. Run the program by going under the Debug menu and click on Start With (or Without) Debugging.

5. Write down the results of the program.
Click on the button several times. What happens? After you have clicked several times, write the message here:

6. To complete the lab, you must PRINT this program. This will be just the Form1.vb tab that contains the code.
When you are done, be sure to save your project to your folder on the H: drive.

3. Testing Arithmetic Operators

For this application, you should create a new project named something like TestArith. This new project should be in the same lab projects folder that you created on your H: drive.

In the **initial form**, change the Text property of the form for the new Test Arithmetic project so that the form box has a nice header label.

This application will be used to test the results of a number of arithmetic operators. The form will be created once with a test program. Then the test program will be continually modified with new operataors to test them.

The form should look something like this:

```
First       |____________________|          | Compute  | (Button1)
Second      |____________________|          | Integers |

Result      |____________________|          |   Clear  | (Button2)
```

It has three Labels, three TextBoxes, and two Buttons.

Section 1: Write the code for testing Integer arithmetic in Buttons 1 and 2

For the Compute Integers button, write the program that

Declares three variables named first, second, and result as type Integer
Get values for first and second from TextBox1 and TextBox2.
Add the two numbers and assign it to the variable result.
Put the result value into TextBox3.

For the Clear button, write the program that

Assigns the empty string “” to all three textboxes

Test for Input Errors:

Start with the operator +, as demonstrated in class. Now run the program and click the Compute Integer button without typing any number. What happens?

Now run the program again, and instead of typing a valid Integer, type a Single – for example, type 3.6 in the first TextBox and type 2.2 in the second TextBox. What happens?

What happens if you type Hello in the first TextBox?

Test the Integer Operators +, -, *, \ : For the remaining tests, put the operator into the program line that computes with the first and second number and assigns it to result. Then run the program with two different sets of numbers and write your results on this paper. For each operator, in the first set of numbers, the first number should be bigger than the second, but in the second set of numbers, the first number should be smaller than the second. Negative numbers can be used. The first two sets of numbers are filled in for you – you can pick the remaining ones to test.

Integer Operator	First Integer	Second Integer	Result
+	9	5	
+	-5	9	
-			
-			
*			
*			
\			
\			
Mod			

Extra: What happens if you use / on Integers such as 11 / 3?

Section 2: Testing Singles

Change the declarations (the lines with Dim) to have type Single instead of Integer.

Test for Input Errors:

Start with the operator + as before. Now run the program and click the Compute Singles button without typing any number. What happens?

Now run the program again, and instead of typing a valid Single, type an Integer – for example, type 6 in the first TextBox and 2 in the second TextBox. What happens?

What happens if you type Hello in the first TextBox?

Test the Single Operators +, -, *, /: For the remaining tests, put the operator into the program line that computes with the first and second number and assigns it to result. Then run the program with two different sets of numbers and write your results on this paper. For each operator, in the first set of numbers, the first number should be bigger than the second, but in the second set of numbers, the first number should be smaller than the second, except for the Exponent operator.

Single Operator	First Single	Second Single	Result
+	2.5	5.0	
+	-5.0	9.655	
-			
-			
*			
*			
/			
/			
^	4.0	2	
^	4.0	3	

When you have completed your testing, hand in this lab paper, along with the printout of the Counting program.