COEN 44 Applied Programming **Fall 2016**

LAB 2: *Writing Your Second Simple C program*

Objectives:

1. To become familiar with vs2015**,** and compile and run a simple C program.

2. To write a simple C program.

3. To learn to use the debugging facilities in Visual Studio C

Part 1: Creating a C program

1. Create a program framework to compile and execute some simple C statements.

```
/*
Date: <Enter the due date here>
Name: <Enter your name here>
Lab #1: Introduction to Visual C
Description:  <Enter a description of what this program
does>
*/
/* Library header files */
#include <stdio.h>// Needed for printf()
#include <stdlib.h>// Needed for system()
int main(void)
{
//YOUR CODE
system("pause");
return 0;
}
```

You should enter the framework above into vs2015 and compile it. When it compiles without errors or warnings, make a copy of the source file then go to step 2.

2. Try the following and write down the results

a. Remove the ";" following *system("pause")* Recompile and note the error that occurs.

b. Replace the ";" you removed, and now put two slashes "//" in front of the #include <stdlib.h> statement. Recompile and observe the error that occurs.

c. Remove the two slashes you inserted, and place them in front of the *return 0;* statement. Recompile and observe the result.

d. Remove the two slashes from c, and put them in front of the *system("pause");* statement and observe what happens when the program compiles and runs.

Part 2: Writing a simple C program.

1. Replace the //YOUR CODE comment in the program frame from Part 1with the following C code

```c
int   width, length, perimeter, area; // integer variables
float  ratio;                         // real (decimal pt) variables

printf("\nCalculations on a rectangle . . .\n\n");
printf("  Enter rectangle width> ");
scanf_s("%d", &width);
printf("  Enter rectangle length>  ");
scanf_s("%d", &length);

perimeter = 2 * width + 2 * length;
area = width * length;
ratio = area / perimeter;

printf("\nPerimeter of rectangle = %d\n\n", perimeter);
printf("Area of rectangle = %d\n\n", area);
printf("Ratio of Area to Perimeter (A/P) = %8.2f\n\n",
ratio);
```

Show the TA that your program executes correctly. Make a copy of the source code from part 2.

2. Try the following and write down the results

   a. Remove the “;” at the end of the line: perimeter = … ; Recompile the program. What happened? Put the “;” (semicolon) you removed back;

   b. Place a // in front of the line: float ratio; Recompile the program. What happened? Remove the // .

3. Verify that your restored program is executing correctly.

Part 3: Working with your program in the debugger.

1. Place a “Breakpoint” at the first printf line.

2. Single step (STEP OVER) through each line of code.

3. On each line, hover the mouse pointer over the variables in the line and note what happens.

4. After stepping over each line, again hover the mouse pointer over the variables in the program

   a. Note how the values change

   b. Look at the “watch window” at the bottom left pane of the Visual Studio window showing the different variables and observe how their values change.

5. Note the value printed on the screen for “ratio”.

a. Now change the line

```
ratio =area / perimeter;
```

to be

```
ratio = (float) area / (float) perimeter
```

b. Re-run the program and note the difference of the result when the ratio is calculated.

Show the TA that your program executes correctly, and that you are able to correctly use the debugger steps

Lab Submission

Make sure the TA saw your program in part 2 running correctly and saw you successfully using the debugger in part 3. Submit electronic copies of the source code from parts 1 and 2 to the TA. Also submit your observations of the effects of the changes made in parts 1, 2 and 3.