

Trent University
COIS1020H
Lab 2 (Windows Version)

1) Selection Statements and the Debugger

In the first part of this lab, we would like to introduce you to Selection Statements (more specifically IF statements) and the Debugger that is available for you to use in Visual Studio.

- a) Enter the following program (use Lab2_1 for both the Project and Code File names). This program uses an IF statement to compute the postal charge for package delivery (there are various classes of packages based on weight plus a flat charge of \$1.25 for every package). The program prompts the user for *weight* (double) and then computes and output the *cost* (double). There is one syntax error in this program that you will need to fix before it will compile. Be careful as the error message may not be the exact error. Look carefully at the whole line where the error occurs and compare it to similar lines.

Please note that there is a text version of this program named Lab2_1.txt (without the line numbers) available in the Labs folder on BlackBoard for copying and pasting. Also, please be aware that the line numbers in the lab exercise may not be the same as the line numbers you see in Visual Studio.

```
1. using System;
2. public static class Lab2_1
3. {
4.     public static void Main()
5.     {
6.         // declare the variables and constants
7.         const double FLAT = 1.25;
8.         double weight;
9.         double cost, pricePerKg = 0;
10.
11.        // Input the weight
12.        Console.Write("Enter a positive weight of the package => ");
13.        weight = Convert.ToDouble(Console.ReadLine());
14.
15.        // Determine the cost per kilogram
16.        if (weight <= 0)
17.            Console.WriteLine("*** Invalid weight");
18.        else if (weight < 1)
19.            pricePerKg = 0.25;
20.        else if (weight <= 3.5)
21.            pricePerKg = 0.5;
22.        else (weight > 3.5)
23.            pricePerKg = 1.0;
24.
25.        // Compute the cost to send the package
26.        if (weight > 0)
27.            cost = weight * pricePerKg + FLAT;
28.        else
29.            cost = 0;
30.        // Output the results
31.        Console.WriteLine("The cost to send the package is {0:C}", cost);
```

```
32.         Console.ReadLine();  
33.     }  
34. }
```

- b) Run the program using an input weight of 2.5

What is the syntax error and how did you fix it?

What is the output?

- c) Run the program using an input weight of 4

What is the output?

- d) Run the program using an input weight of 0.5

What is the output?

- e) Run the program using an input weight of -1

What is the output?

Why do you think the input values for weight in Parts (b)-(e) were chosen?

- f) Now we are going to introduce a very helpful set of tools for finding semantic (logic) errors (i.e., your program compiles but produces the wrong answer). The first debugging tool is the Breakpoint (F9). A Breakpoint causes a program to stop at an executable statement. Let's put in a Breakpoint at Line 16: `if (weight < 0)`. Move your cursor to that line and press (F9). You will see that the line is highlighted in red with a red dot to the left of the line.

Now run the program (F5) using an weight input of 2.5. Notice that the program stops at the line with the Breakpoint (and the line is now highlighted in yellow). Also notice that in the bottom left part of Visual Studio there is a window labeled *Locals* that shows the current values for all the variables declared in `Main()`. Use (F11) to single step your way through the program (paying notice to how the values in the *Locals* window change and which line is highlighted yellow).

Describe what you are seeing as you single step through the program.

- g) Repeat Part (f) for weight inputs of 4, 0.5, and -1.

Comment on what you see.

Answer all the highlighted questions in a file and then submit a PDF of this file (called it Lab2_1.pdf) to the Lab 2 dropbox. When asked "What is the output", simply type in what is seen in the output window.

2) Repetition Statements

- a) Here is a program that uses a sentinel value loop to compute the average for a list of numbers inputted by a user. Enter the following program into C# (ignore the line numbers).

Please note that there is a text version of this program named Lab2_2.txt (without the line numbers) available in the Labs folder on BlackBoard for copying and pasting. Also, please be aware that the line numbers in the lab exercise may not be the same as the line numbers you see in Visual Studio.

```
1. using System;
2. public static class Lab2_2
3. {
4.     public static void Main()
5.     {
6.         int value=0, sum = 0, count = 0;
7.         double average = 0.0;
8.
9.         // Read initial value (seed the loop)
10.        Console.Write("Enter a positive integer (-1 to stop): ");
11.        value = Convert.ToInt32(Console.ReadLine());
12.
13.        // if the inputted value is not the sentinel value, process it
14.        while (value >= 0)
15.        {
16.            // Calculate the running total
17.            sum += value;           // same as sum = sum + value;
18.            // Keep track of the number of inputted values
19.            count++;               // same as count = count + 1;
20.
21.            // Read next value
22.            Console.Write("Enter a positive integer (-1 to stop): ");
23.            value = Convert.ToInt32(Console.ReadLine());
24.        }
25.
26.        // Calculate average (only if the user inputted any numbers
27.        if (count > 0)
28.            average = sum / (double) count;
29.        else
30.            average = 0;
31.
32.        // Print results
33.        Console.WriteLine("sum = {0}, count = {1}", sum, count);
34.        Console.WriteLine("average = {0:F2} ", average);
35.        Console.ReadLine();
36.    }
37. }
```

- b) Build the solution. Resolve any syntax mistakes (if you make any) until the program compiles. Run the program using the input data: 6, 9, 7, 3, -1

What is the output?

- c) Change Line 28 to: `average = sum / count;` and run the program using: 6, 9, 7, 3, -1

What is the output?

How is it different from Part (b), and why?

- d) Change Line 28 to: `average = sum / (1.0 * count);` and run the program using: 6, 9, 7, 3, -1

What is the output?

How is it different from Part (c), and why?

- e) Create your own code to add to the original program. After Line 30, add program code to display whether the number of values inputted (count) is small (< 5) or large (>= 5). Use an If/Else statement.

Show just the If/Else statement.

Run the program using the input 6, 9, 7, 3, -1

What is the output?

- f) Run the program using the input 6, 9, 7, 3, 8, -1

What is the output?

- g) Let's experiment with Breakpoints and loops. Remove the IF statement you added in Part (e) and put in a Breakpoint (F9) at Line 14: `while (value >= 0)`. Run the program with the input: 9, 5, -1. Be sure to single step through (F11) so you can see the variables in the *Locals* window change.

Comment on what you saw.

- h) Repeat Part (g) with the input: -1.

Comment on what you saw.

- i) Leaving the Breakpoint in at Line 14, add a semicolon to the end of the line (i.e.,
`while (value >= 0);`

Run the program with the input: 9, 5, -1. Be sure to single step through (F11).

How does it differ from Part (f) and why?

- j) One last experiment with this program. Remove the semicolon from the end of Line 14. Let's see if it is possible to simplify the program by eliminating the initial prompt (`Console.Write / Console.ReadLine`). Remove Lines 9-24 inclusive and replace with the following:

```
while (value >= 0)
{
    // prompt the user to enter an integer
    Console.Write("Enter a positive integer (-1 to stop): ");
    value = Convert.ToInt32(Console.ReadLine());

    // Calculate the running total
    sum += value;           // same as sum = sum + value;
    // Keep track of the number of inputted values
    count++;               // same as count = count + 1;
}
```

Run the program with the input data: 6, 9, 7, 3, -1

What is the output, and if it's different from Part (d), why?

Answer all the highlighted questions in a file and then submit a PDF of this file (called it Lab2_2.pdf) to the Lab 2 dropbox. When asked "What is the output", simply type in what is seen in the output window.

3) Putting it All Together

For the last part of the lab, you are required to complete the following program that uses a sentinel value loop and if statement to input marks between 0 and 100, determine if the mark is a pass or fail, and output the percentage of passing and failing marks. The program is terminated by entering a negative mark. Enter the following program into C# (ignore the line numbers) and replace the lines marked with `// ***` with the appropriate C# code (may require more than one line of code to complete the missing parts).

Please note that there is a text version of this program named Lab2_3.txt (without the line numbers) available in the Labs folder on BlackBoard for copying and pasting. Also, please be aware that the line numbers in the lab exercise may not be the same as the line numbers you see in Visual Studio.

```
1. // Name: xxxxxxxxxxxxxx
2. // Student Number: xxxxxxxx
3. // Lab 2, Part 3
4. // Program Description: This program uses a loop and if statement to input marks
5. //   between 0 and 100) from a user, determines if the mark is pass or fail
6. //   and outputs the percentage of passing and failing marks. The user then
7. //   terminates the program by entering a negative value for a mark.
8.
9. using System;
10. public static class Lab2_3
11. {
12.     public static void Main()
13.     {
14.         const int PASS = 50;
15.         int numPass=0, numFail=0, totalMarks=0;
```

```

16.     double mark, perPass=0, perFail=0;
17.
18.     // loop to read in a valid mark or the sentinel value
19.     do
20.     {
21.         // Read initial mark (seed the loop)
22.         Console.WriteLine("Enter a mark between 0 and 100 (-ve value to stop): ");
23.         mark = Convert.ToDouble(Console.ReadLine());
24.     } while (mark > 100);
25.     // if the inputted mark is not the sentinel value, process it
26.     while (mark >= 0)
27.     {
28.
29.         // increment the counter for the total number of data values
30.         // *** Insert code
31.
32.         // Determine if the mark is a pass or fail (If statement)
33.         // *** Insert code
34.
35.         // Read next mark
36.         // *** Insert code
37.     }
38.
39.     // Calculate the percentage of marks that were passes and fails
40.     // *** Insert code
41.
42.     // Print results
43.     Console.WriteLine("Total number of marks = {0}", totalMarks);
44.     Console.WriteLine("Percentage of passing marks = {0:P1}", perPass);
45.     Console.WriteLine("Percentage of passing marks = {0:P1}", perFail);
46.     Console.ReadLine();
47. }
48. }

```

Once you are comfortable that the program works correctly, demonstrate it to the lab personnel, and then submit your Lab2_3.cs and Lab2_3.exe files to the Lab 2 dropbox