Trent University
COIS1020H
Lab 2 Answer Document

1) Selection Statements and the Debugger

b) Run the program using an input weight of 2.5

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

'else' doesn't really need parenthesis because everything greater than 3.5 is already included since it was not included in 'if' and 'else if' and so the code expects a semi colon suggesting the end of the 'else' function.

Remove parenthesis

What is the output?

```
Enter a positive weight of the package => 2.5
The cost to send the package is $2.50
```

c) Run the program using an input weight of 4

What is the output?

```
Enter a positive weight of the package => 4
The cost to send the package is $5.25
```

d) Run the program using an input weight of 0.5

What is the output?

```
Enter a positive weight of the package => 0.5
The cost to send the package is $1.38
```

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

What is the output?

```
Enter a positive weight of the package => -1
*** Invalid weight
The cost to send the package is $0.00
```

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

There are different conditions for pricePerKg depending on the weight declared in the code using 'if-else' statements and in order to check if the program is working correctly and that the calculated cost at the end is correct, the weights in parts b-e were chosen that varied

and fulfilled all the conditions in 'if-else' statements.

f) Now run the program (F5) using an weight input of 2.5.

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

When we press f5 and enter 2.5
The values stored in the variables are:-

Name	Value	Type
Weight	2.5	double
Cost	0	double
pricePerKg	0	double
FLAT	1.25	double

The compiler starts checking the if else conditions.
`if (weight <= 0),` and as it compiles the condition isn't satisfied and so it goes to next statement, `else if (weight < 1),` but not satisfied so next `else if (weight <= 3.5),` and since this condition is satisfied it enters this condition to know what must be done, that is, pricePerKg = 0.5; so now pricePerKg has been assigned value equal to 0.5.
The new Values of the variables now.
Now the highlight goes to

Name	Value	Type
weight	2.5	double
cost	0	double
pricePerKg	0.5	double
FLAT	1.25	double

`if (weight > 0),` which is satisfies so it sees what to do when satisfied and that is,
`cost = weight * pricePerKg + FLAT;` so the variable cost has been assigned a value that will be calculated according to the formula, where weight is he input by user and pricePerKg is what was assigned previously 0.5, flat has been already declared as constant 1.5 the new values are -

Name	Value	Type
weight	2.5	double
cost	2.5	double
pricePerKg	0.5	double
FLAT	1.25	double

now output with cost as the value to be printed and C stands for $ to be used

```
Console.WriteLine("The cost to
send the package is {0:C}", cost);
```

Then it reads

```
Console.ReadLine();
```

Indicating to keep the output window open unless user presses a key

Output-
Enter a positive weight of the package => 2.5
The cost to send the package is $2.50

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

`Weight=4`

When we press f5 and enter 4

The values stored in the variables are:-

Name	Value	Type
weight	4	double
cost	0	double
pricePerKg	0	double
FLAT	1.25	double

The compiler starts checking the if else conditions.
`if (weight <= 0),` and as it compiles the condition isn't satisfied and so it goes to next statement, `else if (weight < 1),` but not satisfied so next `else if (weight <= 3.5), not satisfied, so it goes in else` and pricePerKg = 1; so now pricePerKg has been assigned value equal to 1.
The new Values of the variables now.
Now the highlight goes to

Name	Value	Type
weight	4	double
cost	0	double
pricePerKg	1	double
FLAT	1.25	double

`if (weight > 0),` which is satisfies so it sees what to do when satisfied and that is,
`cost = weight * pricePerKg + FLAT;` so the variable cost has been assigned a value that will be calculated according to the formula, where weight is he input by user and pricePerKg is what was assigned previously 1, flat has been already declared as constant 1.25 the new values are -

Name	Value	Type
weight	4	double
cost	5.25	double
pricePerKg	1	double
FLAT	1.25	double

now output with cost as the value to be printed and C stands for $ to be used

```
Console.WriteLine("The cost to
send the package is {0:C}", cost);
```

Then it reads

```
Console.ReadLine();
```

Indicating to keep the output window open unless user presses a key

Output-
Enter a positive weight of the package => 4
The cost to send the package is $5.25

Comment on what you see.

```
Weight=0.5
```

Name	Value	Type
weight	0.5	double
cost	0	double
pricePerKg	0	double
FLAT	1.25	double

When we press f5 and enter 0.5

The values stored in the variables are:-

The compiler starts checking the if else conditions.
`if (weight <= 0),` and as it compiles the condition isn't satisfied and so it goes to next statement, `else if (weight < 1),` satisfied and pricePerKg = 0.25; so now pricePerKg has been assigned value equal to 0.25.
The new Values of the variables now.
Now the highlight goes to

Name	Value	Type
weight	0.5	double
cost	0	double
pricePerKg	0.25	double
FLAT	1.25	double

`if (weight > 0),` which is satisfies so it sees what to do when satisfied and that is,
`cost = weight * pricePerKg + FLAT;` so the variable cost has been assigned a value that will be calculated according to the formula, where weight is he input by user and pricePerKg is what was assigned previously 0.25, flat has been already declared as constant 1.25 the new values are -

Name	Value	Type
weight	0.5	double
cost	1.375	double
pricePerKg	0.25	double
FLAT	1.25	double

now output with cost as the value to be printed and C stands for $ to be used

```
Console.WriteLine("The cost to
send the package is {0:C}", cost);
```

Then it reads

```
Console.ReadLine();
```

Indicating to keep the output window open unless user presses a key

Output-
Enter a positive weight of the package => 0.5
The cost to send the package is $1.375

```
Weight=-1
```

When we press f5 and enter -1

The values stored in the variables are:-

Name	Value	Type
weight	-1	double
cost	0	double
pricePerKg	0	double
FLAT	1.25	double

The compiler starts checking the if else conditions.
`if (weight <= 0),` and as it compiles the condition is satisfied and so output is ***invalid weight

`if (weight > 0),` which is not satisfied so it goes in else which says cost=0.00

Name	Value	Type
weight	-1	double
cost	0	double
pricePerKg	0	double
FLAT	1.25	double

Output-

```
Enter a positive weight of the package => -1
*** Invalid weight
The cost to send the package is $0.00
```

2) Repetition Statements

b) Build the solution. Run the program using the input data: 6, 9, 7, 3, -1

What is the output?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): -1
sum = 25, count = 4
average = 6.25
```

c) Change Line 28 and run the program using: 6, 9, 7, 3, -1

What is the output?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): -1
sum = 25, count = 4
average = 6.00
```

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

We get 6.00 instead of a 6.25 because sum and count both are integers and dividing two integers gives an integer. So when the value is substituted in average it is basically an integer 6 but since average is a double, the output comes out as 6.00 and not 6

d) Change Line 28 and run the program using: 6, 9, 7, 3, -1

What is the output?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): -1
sum = 25, count = 4
average = 6.25
```

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

The final answer of average is 6.25, a double, even when no (double) was used.
That is because we multiplied 1.0 with count and 1.0 is a double so even if count is an int, double multiplied by an int gives a double. So basically, sum is being divided by a double and thus the final quotient is a double. Average =6.25 and not 6.00

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.

```
if(count<5)
    {Console.WriteLine("The number of values inputted is small");}
else
    {Console.WriteLine("The number of values inputted is large");}
```

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

What is the output?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): -1
The number of values inputted is small
sum = 25, count = 4
average = 6.25
```

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

What is the output?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): 8
Enter a positive integer (-1 to stop): -1
The number of values inputted is large
sum = 33, count = 5
average = 6.60
```

g) Let's experiment with Breakpoints and loops. Run the program with the input: 9, 5, -1.

Comment on what you saw.

Enter a positive integer (-1 to stop): 9
is showed on the output window and then we enter the loop since the condition of while(value>=0) Is satisfied

Name	Value	Type
value	9	int
sum	0	int
count	0	int
average	0	double

the next it goes on sum+=value; after clicking F11 the values stored are

Name	Value	Type
value	9	int
sum	9	int
count	0	int
average	0	double

then count++; again after clicking F11 the values stored are

Name	Value	Type
value	9	int
sum	9	int
count	1	int
average	0	double

Now

```
Console.Write("Enter a positive integer (-1 to stop): ");
 value = Convert.ToInt32(Console.ReadLine());
```

we get the output window and enter 5
the new values are
The loop is enter
Condition of while(value>=0) checked and satisfied so
Pressed F11
Sum+=value; highlighted and values stored are

Name	Value	Type
value	5	int
sum	14	int
count	1	int
average	0	double

Name	Value	Type
value	5	int
sum	9	int
count	1	int
average	0	double

Then count++; is highlighted and new values are

Name	Value	Type
value	5	int
sum	14	int
count	2	int
average	0	double

Now

```
Console.Write("Enter a positive integer (-1 to stop): ");
value = Convert.ToInt32(Console.ReadLine());
```

we get the output window and enter -1

the new values are

Name	Value	Type
value	-1	int
sum	14	int
count	2	int
average	0	double

It checks while(value>=0) which is not satisfied so it exits loop and

if (count > 0) is highlighted

which is satisfied and so it enters the if condition and sees what is to be done

```
average = sum / (1.0 * count);
```

Name	Value	Type
value	-1	int
sum	14	int
count	2	int
average	7	double

It goes to next statements and these are highlighted line by line

```
Console.WriteLine("sum = {0}, count = {1}", sum, count);
Console.WriteLine("average = {0:F2} ", average);
Console.ReadLine();
```

And output is showed

```
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 5
Enter a positive integer (-1 to stop): -1
sum = 14, count = 2
average = 7.00
```

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

Comment on what you saw.

Name	Value	Type
value	-1	int
sum	0	int
count	0	int
average	0	double

These are the values stored.
While(value>=0) is not satisfied so it never enters the loop and goes to if else condition
If(count>0) is highlighted and is not satisfied so it enters the else condition

```
average = 0;
```

all the values stored are still the same.

then these statements are highlighted one by one

```
Console.WriteLine("sum = {0}, count = {1}", sum, count);
Console.WriteLine("average = {0:F2} ", average);
Console.ReadLine();
```

And output is showed

```
Enter a positive integer (-1 to stop): -1
sum = 0, count = 0
average = 0.00
```

.

i) 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?

```
Enter a positive integer (-1 to stop): 9
```

This is shown on output window
Then while(value>=0); is highlighted and that's it,
even when we click enter nothing happens because we are stuck on that line forever because value will always be equal to 9 and that satisfies while(value>=0)

j) One last experiment with this program.

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?

```
Enter a positive integer (-1 to stop): 6
Enter a positive integer (-1 to stop): 9
Enter a positive integer (-1 to stop): 7
Enter a positive integer (-1 to stop): 3
Enter a positive integer (-1 to stop): -1
sum = 24, count = 5
average = 4.80
```

This output is different from part(d) because

The first thing done is checking while(value>=0) and in the start value=0 so it enters the loop and then asks the user to enter a value because

```
Console.Write("Enter a positive integer (-1 to stop): ");
value = Convert.ToInt32(Console.ReadLine());
```

so we enter 6. But since we have already entered the loop, count++ is executed and count=1 now.

This is continued till we enter 9,7

Now when we input 3, again since loop condition is satisfied so it enters and executes

```
Console.Write("Enter a positive integer (-1 to stop): ");
value = Convert.ToInt32(Console.ReadLine());
```

here we enter -1

now sum+=value; is executed

sum=25-1=24 which id depicted finally in the final output

and count++ again executed so count=5

Thus when calculations are done after we enter if(count>0) condition because if condition is true so final average is 24/5 and hence we het different answers/output from part(d).

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

Demonstrate to the Lab personnel and Submit the .cs file online