

Questions to be answered

1. Calculator that provides the answer for the profit based on values entered which requires a username and password.

- A. The programmer has decided to name a method / function "InputCostsProcessAndOutputResult()" *still vague!*

This is a poor name for a method due to its length and vague description. This name could be simplified as "Calculation()" Or "CalculatorProcesses", these are short and descriptive without being too specific to what the function is used for, that is what comments are used for in programs.

The programmer which had named these methods may have not given much thought on the structure of the code, the long name could be due to the individual or team rushing to meet a deadline sacrificing the quality of the structure – and in the near future, its readability too.

If the individual is not rushing they may not be that experienced with how they should structure their code, or are not following the company rules for programmers – assuming that their company and team encourage them to structure their code in a specific way which everybody can agree on.

- B. A list of problems to implement the code and attempt to solve them:

1. The username and password will be hard-coded as there is no mention of using databases where the credentials are stored, there is no data being stored from the program either, it is a calculator. A username and password to use a program which calculates profit is unreasonable for the trouble it is worth.
2. If a user enters a selling price which is lower than the cost? This could result in a value that is lower than zero which is accurate if they decide to calculate how much they would lose, but this could easily be a typo or a mix up of the selling price and cost of the product. This could be solved with validation of the entered data.
3. As the user **must** enter the number of products which are sold, and then their values for cost and selling price, the program will either require a FOR loop or for a less resource heavy method, an equation to be calculated in the total which could look like this → $\text{Int total} = (\text{selling} - \text{cost}) * \text{sold};$

The user could enter decimal values or a value lower than zero for the products sold which will skew the total calculation up. Data validation will correct this issue using IF statements and a TryParse C# .Net 4.0+ provided method.

4. When the number of attempts are exceeded, how will the user be reported to the administrator? Will it be via the network? Will the program require an internet connection or their user account on machine to be connected to the network? The user could easily disconnect from the internet, run a brute force client program to continuously run the program until they gain access.



The program will have multiple errors if it were not tested, these are the steps to break the program to prevent others from doing so:

- Then entering no username and no password when requested by giving no information / data.

- [illegible]

- A decimal value. (Ex. 54.432)
- A value below zero (Ex. -61)
- A value which is zero. (Ex. 0)
- A value above the maximum byte size of a decimal variable. (Ex. 999999999999999999)

- A decimal value. (Ex. 14.442)
- A value below zero (Ex. -61)
- A value which is zero. (Ex. 0)
- A value above the maximum byte size of a decimal variable. (Ex. 99999999999999999999)

The decimal value should be fine as the product variable-type is either a float or double. If the user enters a value below the cost price the program should prompt the user to enter a value above the cost price.

specific data with correct results
not mentioned

7

2. Body Mass Index (BMI) program.

Firstly the user will be prompted for their name, weight and height which will be validated after they have entered, for example the user may enter a value less than zero for their height and weight which would not be correct and can crash the program if the BMI is calculated where zero is divided by zero.

If the user enters an inappropriate or an unrealistically weight and height they will be prompted to enter their credentials again within a WHILE LOOP.

The user could enter nothing for their name or enter a number within their name which would not be correct and this would result in incorrect data, to prevent this a WHILE loop until the error is corrected would be beneficial as an IF statement is only read once and not repeated.

This is pseudo-code which explains the whole program in more detail:

```
[START]
    DECIMAL BMI
    TEXT name
    DECIMAL weight
    DECIMAL height
    ASK USER name
    WHILE USER name CONTAINS A NUMBER(S) –
        PRINT "Do not enter a number please."
        ASK USER name
    –:
    ASK USER weight
    WHILE THE USERS weight IS LESS THAN ZERO –
        PRINT "Do not enter a value below zero, please."
        ASK USER weight
    –:
    WHILE USER weight IS ABOVE 300 –
        PRINT "Nobody is that fat and their BMI would be the last thing to be
        worrying about."
        ASK USER weight
    –:
    ASK USER height
    WHILE USER height IS LESS THAN ZERO –
        PRINT "Do not enter a value below zero, please."
        ASK USER weight
    –:
    WHILE USER height IS ABOVE 2.1 –
        PRINT "Please enter an appropriate value for your height."
        ASK USER height
    –:
```

BMI EQUALS weight / (height * height)
IF BMI IS LESS THAN 18.5 –
PRINT "You are underweight, [name]"

$$BMI = \frac{wt}{h^2}$$

–:
IF BMI IS LESS THAN 18.5 –
PRINT "You are underweight, [name]."
–:

IF BMI IS GREATER THAN 18.5 AND LESS THAN 25 –
PRINT "You are not underweight and not overweight, [name]."
–:

IF BMI IS GREATER THAN 25 AND LESS THAN 30 –
PRINT "You are overweight, [name]"
–:
IF BMI IS GREATER THAN 30 –

PRINT "You are obese, go outside for a walk everyday and drink
2.5Litres of water and avoid eating junk food that raises your blood
pressure (usually contains lots of salt and sugar), [name]."
–:

[END]

18

3. Appropriate functions are used for a menu where the user can enter values to calculate the area of the rectangle and the circle.

A. A list of problems within this code:

1. The first issue of this program is the user entering an option in the menu which is not available, there are only three options to choose from, the number one to three.

This can be solved with validation using an IF statement for the user entering a number which is NOT 1 OR NOT 2 OR NOT 3. Using an ELSE statement would run the code within it when the menu is being drawn after running through the methods which is not ideal.

2. After the menu and the methods have been ran through, it looks as if the menu is contained within a WHILE loop as there is an option to exit which implies that the whole program cannot break / exit on its own, the user must provide input for this to occur. Using methods are appropriate due to keeping code readable among other reasons for the fact it can be reused multiple times within the program if required.

3. When the user decides to enter values for the rectangle and circle they should not enter values such as:

- Values below zero (-324)
- Values above the maximum byte size of a FLOAT or DOUBLE.
- Alphabetical characters.

If the user decides to enter inappropriate data there will be a WHILE loop using a TryParse method to validate data after it is entered, a Boolean variable will be used to avoid the incorrect values from passing through the code to a crash.

4. When the user decides to exit the program with the number "3" the user is actually exiting the loop (assuming there is nothing more beyond the while loop) which will allow for the program to end itself, if there is anything beyond that point this could result in unexpected programmed content being shown.

IS

What about details of calculations?

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  //Dewi
7  namespace CalcuatlionofShapesz
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             float width, height, radius;
14             char userChoice = '0';
15             while (true)
16             {
17                 Menu();
18                 Console.WriteLine("Enter an option: "); userChoice =
19                     Console.ReadKey().KeyChar;
20                 Console.Clear();
21                 if (userChoice == '1')
22                 {
23                     Rectangle(out width, out height);
24                     Console.WriteLine($"The area of a rectangle is: {width *
25                         height}");
26                 }
27                 if (userChoice == '2')
28                 {
29                     Circle(out radius);
30                     Console.WriteLine($"The area of a circle is: {Math.PI *
31                         radius}");
32                 }
33                 if (userChoice == '3')
34                 {
35                     Console.WriteLine("Program is exiting...");
36                     Console.ReadKey();
37                     break;
38                 }
39                 if (userChoice != '1' || userChoice != '2' || userChoice != '3')
40                 {
41                     Console.WriteLine("Please enter an appropriate option,
42                         please.");
43                 }
44             }
45         }
46
47         static void Menu()
48         {
49             Console.WriteLine("GRI19845230 - Dewi Griffiths - Mock Assessment.");
50             Console.WriteLine("-----MENU-----");
51             Console.WriteLine("1) Calculate area of a rectangle");
52             Console.WriteLine("2) Calculate area of a circle");
53             Console.WriteLine("3) Exit program.");
54             Console.WriteLine("-----");
55         }
56
57         static float Rectangle(out float w, out float h)
58         {
59             w = 0f; h = 0f;
60             bool check = false;
61
62             while (!check)
63             {
64                 Console.Write("Please enter the width of the rectangle: ");
65                 check = false;
66                 check = float.TryParse(Console.ReadLine(), out w);
67                 if (check == false)
68                 {
69                     Console.WriteLine("Please enter appropriate values please.");
70                 }
71                 else if (w <= 0)
72                 {

```

Q2 - 0

Q3 - 25

```

68         Console.WriteLine("Do not enter a value below zero please,
69         thanks.");
70         check = false;
71     }
72     Console.WriteLine("Press any key to continue...");
73     Console.ReadKey();
74     Console.Clear();
75 }
76 Console.Clear();
77 check = false;
78
79 while (!check)
80 {
81     Console.Write("Please enter the height of the rectangle: ");
82     check = false;
83     check = float.TryParse(Console.ReadLine(), out h);
84     if (check == false)
85     {
86         Console.WriteLine("Please enter appropriate values please.");
87     }
88     else if (h <= 0)
89     {
90         Console.WriteLine("Do not enter a value below zero please,
91         thanks.");
92         check = false;
93     }
94     Console.WriteLine("Press any key to continue...");
95     Console.ReadKey();
96     Console.Clear();
97 }
98 return w;
99 }
100 static float Circle(out float r)
101 {
102     r = 0f;
103     bool check = false;
104
105     while (!check)
106     {
107         Console.Write("Please enter the radius of the circle: ");
108         check = float.TryParse(Console.ReadLine(), out r);
109         if (check == false)
110         {
111             Console.WriteLine("Please enter appropriate values please.");
112         }
113         else if (r <= 0)
114         {
115             Console.WriteLine("Please enter a value above zero please,
116             thank you.");
117         }
118         Console.WriteLine("Press any key to continue...");
119         Console.ReadKey();
120         Console.Clear();
121     }
122     return r;
123 }
124 }
125

```

Handwritten notes:

- Next to line 98: with formula
- Next to line 122: why return r?