

- A. A programmer has created a function named *InputCostsProcessAndOutputResult*. Give reasons why this function name is a poor choice and why it suggests the programmer has not given sufficient attention to the structure of the code.

The function name does not adequately describe the action being performed by the function (badly named).

The name indicates that the function is performing multiple actions and that the programmer has failed to adequately evaluate the problem.

[4 marks]

- B. Provide a list of the problems that must be solved to implement the code for this application and place them in the order that you would attempt to solve them. Avoid vague statements that may involve solving multiple problems. However, the problems identified may be posed in the form of questions e.g. "How do I choose a colour?".

Answers will vary but should be similar to the following:

- Ask user to login
- Store correct username and password
- Read user name and password ("hide" password?)
- Count the attempts
- Start a timer 3 bad attempts are made
- Enable and disable login
- Display prompt for user to enter price
- Input number
- Check valid number entered
- Retry number entry
- Input number of products
- Store multiple prices
- Enter multiple product prices
- Calculate tax paid on product
- Calculate total tax paid
- Display total tax paid

[8 marks]

- C. Describe the tests you would use to confirm the program is running correctly.

Check login with correct username and password
Check login with incorrect username
Check login with incorrect password
Check login disabled after 3 attempts
Check login re-enabled after 5 minutes
Check error indicated when invalid number entered i.e. "text"
Check number is re-entered until valid value provided

[8 marks]

1. Body Mass Index (BMI) is calculated by using the formula

$$\text{BMI} = \text{kg} / \text{m}^2$$

where kg is the weight in kilograms and m is the height in metres.

[ASK is you do not understand this formula.]

A program is needed ask the user to enter a name, weight and height and then display a BMI Description based upon the calculated BMI value and the following grid.

BMI Value	Description
Less than 18.5	Underweight
Greater than 18.5 and less than 25	Normal
Greater than 25 and less than 30	Overweight
30 and above	Obese

A.

Explain the steps and the logic required to solve this problem.

Prompt for weight

Validate weight input is numeric (floating point), sensible range (e.g. 50 – 300 kg)

Prompt for height

Validate height input is numeric (floating point), sensible range (e.g. 1.0 – 2.5m)

Calculate BMI = weight / height squared

If BMI < 18.5 Display “underweight”

If BMI between 18.5 and 25 Display “normal”

If BMI between 25 and 30 Display “overweight”

If BMI >30 Display “obese”

[20 marks]

B.

Implement the code for your solution.

[10 marks]

```

namespace Q28BMI
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("BMI Calculator");
            bool OK = false;
            double weight = 0;
            double height = 0;
            Console.Write("Please enter your name:");
            string name = Console.ReadLine();
            while (!OK)
            {
                Console.Write("Enter weight in kilograms:");
                OK = double.TryParse(Console.ReadLine(), out weight);
            }
            OK = false;
            while (!OK)
            {
                Console.Write("Enter height in metres:");
                OK = double.TryParse(Console.ReadLine(), out height);
            }

            double BMI = weight / (height * height);
            if (BMI < 18.5)
            {
                Console.WriteLine("{0}, you are underweight", name);
            }
            if (BMI >= 18.5 && BMI < 25)
            {
                Console.WriteLine("{0}, you are normal", name);
            }
            if (BMI >= 25 && BMI < 30)
            {
                Console.WriteLine("{0}, you are Overweight", name);
            }
            if (BMI >= 30)
            {
                Console.WriteLine("{0}, you are Obese", name);
            }
        }
    }
}

```

2. Using **appropriate functions** implement an application which displays a text-based menu listing that allows the user to select between calculating the area of a rectangle, calculating the area of a circle and exiting. For example, the menu might display “1. Calculate area of rectangle” and the user must enter the number one to select this option. If they choose not to exit then the menu should be redisplayed and they should be allowed to select another option. When calculating the area of the rectangle they must be prompted to enter the width and the height of the rectangle, but when calculating the area of a circle they should be prompted to enter the radius. In both cases, the result should be displayed once it has been calculated.

The program may look like this:

```
1. Calculate area of rectangle
2. Calculate area of circle
3. Exit

Choose option: 1          ← Text shown in RED is entered by user
Please enter width: 2
Please enter height: 3
Area of rectangle is 6

Choose option: 3
Program is exiting
```

- A. Provide a list of the problems that must be solved to implement the code for this application and place them in the order that you would attempt to solve them. Avoid vague statements that may involve solving multiple problems. However, the problems identified may be posed in the form of questions e.g. “How do I choose a colour?”.

[20 marks]

- B. Write the code for the functions required to solve the problems previously identified.

Hint 1: Develop the code in stages, marks will be awarded for any working code that contributes to the solution.

Hint 2: Area of rectangle = width x height

Hint 3: Area of circle = $\text{PI} * \text{radius}^2$ ← Ask if you don't understand “squared”

[30 marks]

```

namespace Q2Area
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Areas");
            string option = "";
            while (option != "3")
            {
                Menu();
                option = Console.ReadLine();
                if (option == "1")
                {
                    Rectangle();
                }
                if (option == "2")
                {
                    Circle();
                }
            }
        }
    }
}

```

```

static void Menu()
{
    Console.WriteLine("Menu");
    Console.WriteLine("1. Rectangle");
    Console.WriteLine("2. Circle");
    Console.WriteLine("3. Exit");
    Console.Write("\n\nPlease select a menu choice:");
}

```

```

static void Rectangle()
{
    bool OK = false;
    double length = 0;
    double width = 0;
    while (!OK || length < 0)
    {
        Console.Write("Enter length of rectangle:");
        OK = double.TryParse(Console.ReadLine(), out length);
    }

    OK = false;
    while (!OK || width < 0)
    {
        Console.Write("Enter width of rectangle:");
        OK = double.TryParse(Console.ReadLine(), out width);
    }
    double area = length * width;
    Console.WriteLine("Area is {0}", area);
}

```

```

static void Circle()
{
    bool OK = false;
    double radius = 0;

    while (!OK || radius < 0)
    {
        Console.Write("Enter radius:");
        OK = double.TryParse(Console.ReadLine(), out radius);
    }
    double area = Math.PI * radius * radius;
    Console.WriteLine("Area is {0}", area);
}

```

Diwedd/End