

Task 1

3 The function name "InputCostsProcessAndOutputResult" is a poor choice because the name is very long for a function, this would mean having to type out the long function name whenever the programmer wishes to use it. This function name suggests that the programmer hasn't given sufficient attention to the structure of the code because it isn't split up properly. Inputting the costs and outputting the results should be in separate functions to allow for a more efficient program. |

Login

1. Create string variables called 'Username' and 'Password' and set the username and password that you want.
2. Create a while loop, this loop will contain the login process along with the login validation, this loop allows the user to input the username and password again if they are incorrect.
3. Create an output that asks the user for the username so that they can start logging in.
4. Create a string variable for the users username input.
5. Create an output that asks the user for the password so that they can login.
6. Create another string variable which allows the user to enter what they believe the password is.

Login Validation

1. Create an integer variable called 'Attempts' and set the value to 3, this will be the way to limit the users login attempts.
2. Create an IF statement which checks if the 'Attempts' variable is equal to 0, if the variable isn't equal to 0 then it will move on from this step.
 - a. IF the variable is equal to 0 then it will lock the login process for 5 minutes.
3. Create an IF statement which checks to see if the username matches what the user input for the username.
 - a. IF the username is incorrect then the 'Attempts' variable will -1 and the program will output 'Invalid login details please try again. You have X many attempts remaining'.
 - b. IF the username is correct then it will move to the next IF statement for the password. This IF statement will check if the password the user entered matches the password variable.
 - i. IF the password is incorrect then the 'Attempts' variable will -1 and the program will output 'Invalid login details please try again. You have X many attempts remaining'.
 - ii. IF the password is correct then it will exit the loop and log the user in allowing them to use the program.

Tests

1. Test to see if the attempts go down if the user has incorrect login details.
 - a. This would be tested by entering incorrect login details.
2. Test to see if the program loops around after the login details are incorrect.
 - a. This would be tested by entering incorrect login details.
3. Test to see if the program locks the user from logging in for 5 minutes.
 - a. This would be tested by entering incorrect login details 3 times.
4. Test to see if the program allows the user to login if the details are correct.
 - a. This would be tested by entering the correct login details.

4
Test for correct function?
Test for correct calculation?

Task 2**Menu Creation**

1. Create a loop which will loop back around if the option that the user selects is invalid.
2. Create a menu which asks the user what they would like select. This would include option 1; area of rectangle, option 2; area of circle and option 3; exit program.
3. Allow the user to input which option they would like to select.
4. Create an IF statement which checks if the user has input a valid option (1, 2 or 3)
 - a. IF the user has not then the program will loop back around to allow the user to try again.
 - b. IF the user has entered a valid option then it will go to that one

Option 1 'Calculate area of rectangle'

1. Create a new loop for validation purposes. This will make sure that all user inputs will be valid to allow the calculations to be made.
2. Create three new variables called 'width', 'height' and 'rectangleTotal'.
3. Ask the user what the width of the rectangle is.
4. Use the width variable to allow the user to input a number.
5. Create an IF statement which checks if it is a number and if it is above 0.
 - a. IF the input was invalid then output 'Invalid input please try again' and will loop back around.
 - b. IF the input was valid then the program will move on.
6. Ask the user what the height of the rectangle is.
7. Use the height variable to allow the user to input a number.
8. Create an IF statement which checks if it is a number and if it is above 0.
 - a. IF the input was invalid then output 'Invalid input please try again' and will loop back around.
 - b. IF the input was valid the total will be calculated. 'rectangleTotal = width * height'
9. Output the area of the rectangle for the user.
10. Close the application.

Option 2 'Calculate area of circle'

1. Create a new loop for validation purposes. This will make sure that all user inputs will be valid to allow the calculations to be made.
2. Create three new variables called 'radius' and 'circleTotal'. *three?*
3. Ask the user what the radius of the circle is.
4. Use the radius variable to allow the user to input a number.
5. Create an IF statement which checks if it is a number and if it is above 0.
 - a. IF the input was invalid then output 'Invalid input please try again' and will loop back around.
 - b. IF the input was valid then the program will calculate the total. 'circleTotal = 3.14 * radius²'
6. Output the area of a circle for the user.
7. Close the application

Option 3 'Exit'

1. Exit the program

// Bradley Martin

// MAR16829326

using System;

namespace BradleyMartin_Practical_Assignment_Program

{

class Program

{

static void Main(string[] args)

{

Menu(); // Menu Method

int menuOption;

int loop = 0;

while (loop == 0) // loop validation until correct

{

if (int.TryParse(Console.ReadLine(), out menuOption)) // number validation

{

if (menuOption == 1) // Area of Rectangle option chosen

{

loop++; // close loop

Console.WriteLine("Area of Rectangle Chosen!");

AreaRectangle(); // go to rectangle method

}

else

{

if (menuOption == 2) // Area of Circle option chosen

{

loop++; // close loop

Console.WriteLine("Area of Circle Chosen!");

AreaCircle(); // go to circle method

}

else

{

if (menuOption == 3) // Exit option chosen

{

Environment.Exit(0); // Exit Application

}

else // If the input was a number but an invalid option

{

Console.WriteLine("Please enter a valid option");

}

}

}

else // If the input wasn't a number

{

Console.WriteLine("Please enter a valid option");

}

}

static void Menu() // Menu Creation

{

Console.WriteLine("Please select one of the options below:");

Console.WriteLine("1. Calculate Area of Rectangle");

Console.WriteLine("2. Calculate Area of Circle");

Console.WriteLine("3. Exit Program");

}

static void AreaRectangle() // Area of Rectangle Calculation Method

{

double width = 0; // width variables for calculations

int WidthLoop = 0;

Console.WriteLine("What is the width of the rectangle?");

while (WidthLoop == 0) // validation loop

{

double widthh = 0;

if (double.TryParse(Console.ReadLine(), out widthh)) // number validation

{

Store
way to
exit a
loop?

30

```

        if (widthh > 0) // validation for above 0
        {
            WidthLoop++;
            width = widthh;
        }
        else
        {
            Console.WriteLine("Please enter a number above 0!"); //
            validation for inputs for numbers below 1
        }
    }
    else
    {
        Console.WriteLine("Please enter a valid number!"); // validation for
        inputs that aren't numbers
    }
}

double height = 0; // height variable for calculations
int HeightLoop = 0;
Console.WriteLine("What is the height of the rectangle?");
while (HeightLoop == 0) // validation loop
{
    double heightt = 0;
    if (double.TryParse(Console.ReadLine(), out heightt)) // number validation
    {
        if (heightt > 0) // validation for above 0
        {
            HeightLoop++; // close loop
            height = heightt;
        }
        else
        {
            Console.WriteLine("Please enter a number above 0!"); //
            validation for inputs for numbers below 1
        }
    }
    else
    {
        Console.WriteLine("Please enter a valid number!"); // validation for
        inputs that aren't numbers
    }
}

double total = width * height; // calculate area
Console.WriteLine("The area of the rectangle is " + total); // output
Console.ReadLine();
Environment.Exit(0); // exit program
}

static void AreaCircle() // Area of Circle Calculation Method
{
    double radius = 0; // radius variable for calculation
    double pi = 3.14; // pi variable for calculation
    int RadiusLoop = 0;
    Console.WriteLine("What is the radius of the circle?");
    while (RadiusLoop == 0) // validation loop
    {
        double radiuss = 0;
        if (double.TryParse(Console.ReadLine(), out radiuss))
        {
            if (radiuss > 0) // validation for numbers above 0
            {
                RadiusLoop++; // close loop
                radius = radiuss * radiuss;
            }
            else
            {

```

```
    Console.WriteLine("Please enter a number above 0!"); //  
    validation for inputs for numbers below 1
```

```
    }  
}
```

```
else
```

```
{
```

```
    Console.WriteLine("Please enter a valid number!"); // validation for  
    inputs that aren't numbers
```

```
}
```

```
}
```

```
double total = pi * radius; // calculate area
```

```
Console.WriteLine("The area of the circle is " + total); // output area
```

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

```
Environment.Exit(0); // exit program
```

```
}
```

```
}
```

```
}
```



Exam Marking ACCA4016 – Academic and Employability Skills 12th March 2019

Student: **Bradley Martin**

	Exam Date: 12 th March 2019	Marking Date: 15 th March 2019
Question	Max Possible Marks	Marks Achieved
1a	4	3 ✓
1b	8	8 ✓
1c	8	4 ✓
2a	20	20 ✓
2b	30	30 ✓
3a	20	0
3b	10	0
Total	100	65 ✓

Lecturer: G.P. Dawkins

