Task 1

Objective: Implement a program that uses function overloading to calculate the area of a rectangle and a square based on user input.

1. Create two overloaded functions named area:
   - One that takes two parameters (length and width) to calculate the area of a rectangle.
   - Another that takes one parameter (side length) to calculate the area of a square.
2. User Input:
   - Prompt the user to choose whether they want to calculate the area of a rectangle or a square.
   - If the user chooses to calculate the area of a rectangle, ask for both the length and width.
   - If the user wants to calculate the area of a square, ask for the side length.
3. Output:
   - Display the calculated area based on the user's input.

Sample Output

```
Choose the shape to calculate the area:
1. Rectangle
2. Square
Enter your choice (1 or 2): 1
Enter the length of the rectangle: 5
Enter the width of the rectangle: 10
Area of the rectangle: 50

Choose the shape to calculate the area:
1. Rectangle
2. Square
Enter your choice (1 or 2): 2
Enter the side length of the square: 4
Area of the square: 16
```

Task 2

Copy and Paste the following code, the solution to previous lab

Extract input logic to a function so that code duplication is avoided.

```
inputScale = getInputScale();
outputScale = getInputScale();
```

```cpp
#include <iostream>
using namespace std;

enum TemperatureScale
{
    Celsius,
    Fahrenheit,
    Kelvin
};

// function declaration
double convertTemperature(double, TemperatureScale, TemperatureScale);

int main()
{
    double temp;
    TemperatureScale inputScale, outputScale;
    bool flag = true;

    do
    {
        cout << "Enter Temperature: ";
        cin >> temp;

        int opt = 0;
        cout << "Enter Input Scale:" << endl;
        cout << "0. Celcius" << endl;
        cout << "1. Fahrenheit" << endl;
        cout << "2. Kelvin" << endl;
        cin >> opt;

        switch (opt)
        {
        case 0:
            inputScale = Celsius;
            break;
        case 1:
            inputScale = Fahrenheit;
            break;
        case 2:
            inputScale = Kelvin;
            break;
        default:
            cout << "Wrong input." << endl;
            continue;
            break;
        }
```

```
        cout << "Enter Output Scale:" << endl;
        cout << "0. Celcius" << endl;
        cout << "1. Fahrenheit" << endl;
        cout << "2. Kelvin" << endl;
        cin >> opt;

        switch (opt)
        {
        case 0:
            outputScale = Celsius;
            break;
        case 1:
            outputScale = Fahrenheit;
            break;
        case 2:
            outputScale = Kelvin;
            break;
        default:
            cout << "Wrong input." << endl;
            continue;
            break;
        }

        cout << "Converted Temperature :" << convertTemperature(temp,
inputScale, outputScale) << endl;

        char choice = 'n';
        cout << "Enter y to continue and n to exit: ";
        cin >> choice;
        if (choice == 'n')
        {
            flag = false;
        }
    } while (flag);

    return 0;
}


// Function to convert temperature between different scales
double convertTemperature(double temperature, TemperatureScale
inputScale, TemperatureScale outputScale)
{
    // Return the original value if input and output scales are the
same
    if (inputScale == outputScale)
    {
        return temperature;
    }
    // Conversion logic based on input and output scales
```

```
    switch (inputScale)
    {
    case Celsius:
        if (outputScale == Fahrenheit)
        {
            return (temperature * 9.0 / 5.0) + 32;
        }
        else
        {
            return temperature + 273.15;
        }
        break;
    case Fahrenheit:
        if (outputScale == Celsius)
        {
            return (temperature - 32) * 5.0 / 9.0;
        }
        else
        {
            return ((temperature - 32) * 5.0 / 9.0) + 273.15;
        }
        break;
    case Kelvin:
        if (outputScale == Celsius)
        {
            return temperature - 273.15;
        }
        else
        {
            return ((temperature - 273.15) * 9.0 / 5.0) + 32;
        }
        break;
    }
}
```

Task 3

Objective: Implement a recursive function to calculate the Fibonacci number at a given position.

Instructions

- Write a function int fibonacci(int n) that returns the n-th Fibonacci number.
- Use the base cases fibonacci(0) = 0 and fibonacci(1) = 1.

The Fibonacci sequence is defined mathematically as follows:

1. **F(0) = 0** (the first Fibonacci number)
2. **F(1) = 1** (the second Fibonacci number)
3. For (n): **F(n) = F(n - 1) + F(n - 2)**

Sample Output

```
Fibonacci number at position 5 is 5
Fibonacci number at position 10 is 55
Fibonacci number at position 15 is 610
```