

## Lab # 2

### Classes

#### Introduction to Classes and Objects

##### Objective:

The objective of this lab is to introduce the concept of classes and objects in C++. By the end of this lab, you should be able to:

1. Define a class with data members and member functions.
2. Create objects of a class.
3. Understand the difference between classes and objects.
4. Understand the concept of access specifiers (public, private, protected).

##### 1. Classes in C++

- A class is a user-defined data type that encapsulates data members (attributes) and member functions (methods).
- It acts as a blueprint for creating objects.

##### Syntax:

```
cpp                                                                    Copy
class ClassName {
private:
    // Data members (attributes)
    // Member functions (methods)
public:
    // Data members (attributes)
    // Member functions (methods)
};
```

##### 2. Objects in C++

- An object is an instance of a class. It is a real-world entity that has state (data members) and behavior (member functions).

##### Syntax:

```
cpp                                                                    Copy
ClassName objectName;
```

### 3. Access Specifiers

- **Public:** Members are accessible from outside the class.
  - **Private:** Members are not accessible from outside the class (default for classes).
  - **Protected:** Similar to private but accessible in derived classes.
- 

## Lab Activities:

### Activity 1: Define a Class and Create Objects

Define a class named Student with the following:

- Data members: name, rollNumber, age
- Member functions: display() to print the student details.

#### Code:

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

class Student {
private:
    string name;
    int rollNumber;
    int age;

public:
    // Member function to set data
    void setData(string n, int r, int a) {
        name = n;
        rollNumber = r;
        age = a;
    }

    // Member function to display data
    void display() {
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNumber << endl;
        cout << "Age: " << age << endl;
    }
};
```

```

int main() {
    // Create objects of the class Student
    Student student1;
    Student student2;

    // Set data for student1
    student1.setData("John Doe", 101, 20);

    // Set data for student2
    student2.setData("Jane Smith", 102, 21);


    // Display student details
    cout << "Student 1 Details:" << endl;
    student1.display();
    cout << endl;

    cout << "Student 2 Details:" << endl;
    student2.display();

    return 0;
}

```

#### Output:



```

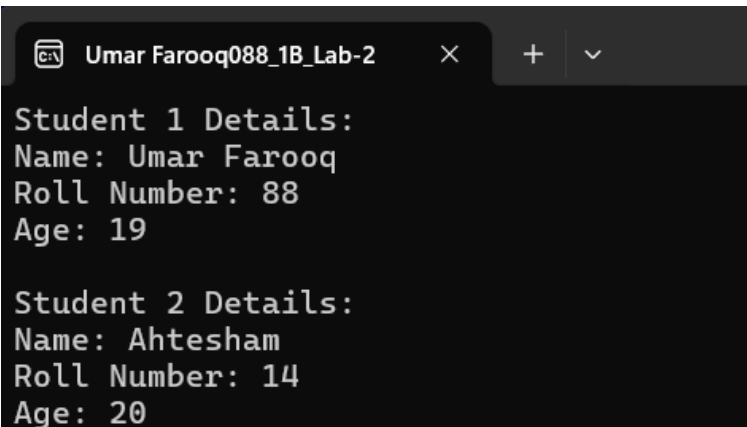
Student 1 Details:
Name: John Doe
Roll Number: 101
Age: 20

Student 2 Details:
Name: Jane Smith
Roll Number: 102
Age: 21

```

#### Your Output Here....

In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab02_A1.exe`



```

Student 1 Details:
Name: Umar Farooq
Roll Number: 88
Age: 19

Student 2 Details:
Name: Ahtesham
Roll Number: 14
Age: 20

```

---

## Activity 2: Access Specifiers

Modify the Student class to demonstrate the use of access specifiers (public, private).

### Code:

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

class Student {
private:
    string name;
    int rollNumber;

public:
    int age; // Public data member

    // Public member function to set private data
    void setData(string n, int r, int a) {
        name = n;
        rollNumber = r;
        age = a;
    }

    // Public member function to display data
    void display() {
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNumber << endl;
        cout << "Age: " << age << endl;
    }
};

int main() {
    Student student1;

    // Set data for student1
    student1.setData("Alice Johnson", 103, 22);

    // Access public data member directly
    cout << "Accessing public data member (age): " << student1.age << endl;

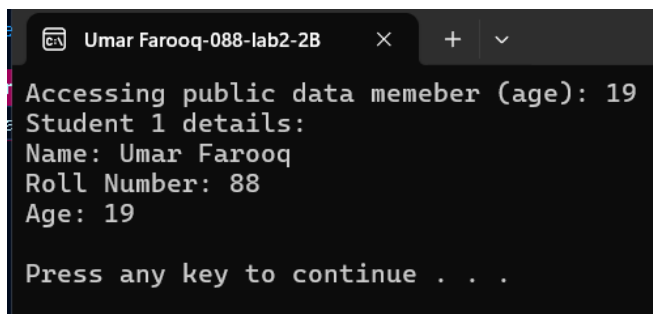
    // Display student details
    cout << "Student 1 Details:" << endl;
    student1.display();
    return 0;}
```

### Output:

```
Accessing public data member (age): 22
Student 1 Details:
Name: Alice Johnson
Roll Number: 103
Age: 22
```

### Your Output Here....

In case of output snippet please make sure output snippet contains student name and id. [AliAhmed\\_123\\_Lab02\\_A2.exe](#)

A screenshot of a terminal window with a dark background. The title bar shows 'Umar Farooq-088-lab2-2B'. The output text is: 'Accessing public data memeber (age): 19', 'Student 1 details:', 'Name: Umar Farooq', 'Roll Number: 88', 'Age: 19', and 'Press any key to continue . . .'.

```
Accessing public data memeber (age): 19
Student 1 details:
Name: Umar Farooq
Roll Number: 88
Age: 19
Press any key to continue . . .
```

### Activity 3: Class with Member Functions Defined Outside

Define a class Rectangle with data members length and width.

Define member functions setData() and calculateArea() outside the class.

#### Code:

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

class Rectangle {
private:
    float length;
    float width;

public:
    void setData(float l, float w); // Function declaration
    float calculateArea();          // Function declaration
};

// Function definitions outside the class
void Rectangle::setData(float l, float w) {
```

```

length = 1;
    width = w;
}

float Rectangle::calculateArea() {
    return length * width;
}

int main() {
    Rectangle rect;

    // Set data for rectangle
    rect.setData(5.5, 4.0);

    // Calculate and display area
    cout << "Area of Rectangle: " << rect.calculateArea() << endl;

    return 0;
}

```

#### Output:

Copy

Area of Rectangle: 22

#### Your Output Here....

In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab02_A3.exe`

Umar Farooq-088-Lab2-1B
✕
+
▼

```

Area of Rectangle: 40.05
Press any key to continue . . .

```

---

#### Conclusion:

In this lab, we learned:

1. How to define a class with data members and member functions.
2. How to create objects of a class and access their members.
3. The importance of access specifiers in encapsulation.
4. How to define member functions outside the class.

---

#### Additional Notes:

- Classes are the foundation of Object-Oriented Programming (OOP) in C++.
- Objects are instances of classes and hold separate copies of data members.
- Access specifiers control the visibility of class members.

## Lab Tasks/ Homework

**Task 01:** Write a program to add two different times (hours, minutes and seconds) using simple class and display total time and convert time into number of days.

Write functions:

```
getTime();  
displayTime(Time obj);  
void addTime(Time obj1, Time obj2);  
int numberOfDays(Time obj);
```

**Solution:**

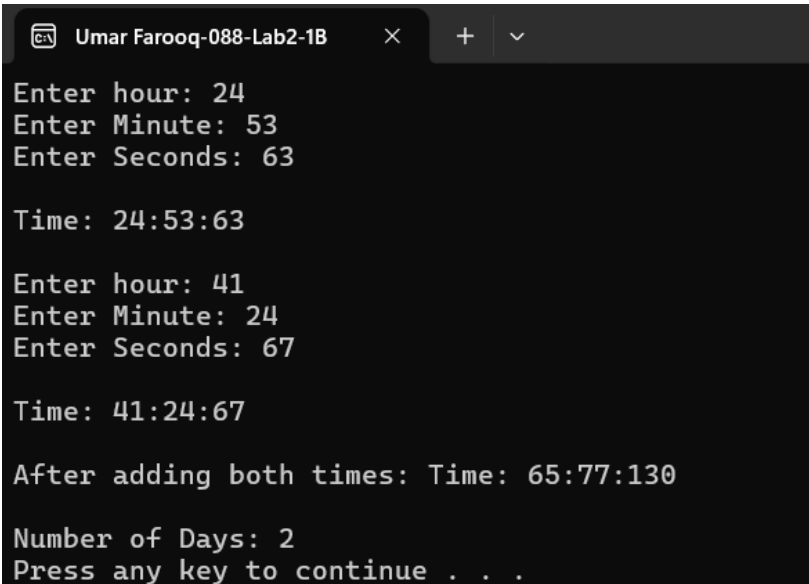
```
#include <iostream>  
#include <cstdlib>  
using namespace std;  
class Time {  
private:  
    int hour;  
    int minute;  
    int seconds;  
public:  
    void setdata(int H, int M, int S)  
    {  
        hour = H;  
        minute = M;  
        seconds = S;  
    }  
    void getTime()  
    {  
        cout << "Enter hour: ";  
        cin >> hour;  
        cout << "Enter Minute: ";  
        cin >> minute;  
        cout << "Enter Seconds: ";  
        cin >> seconds;  
    }  
    void displaytime()  
    {  
        cout<<"Time: " << hour << ":" << minute << ":" << seconds <<  
endl;  
    }  
    void addTime(Time t1, Time t2)  
    {  
        cout << "After adding both times: ";  
        hour = t1.hour + t2.hour;
```

```

        minute = t1.minute + t2.minute;
        seconds = t1.seconds + t2.seconds;
    }
    int numberOfDays()
    {
        return hour / 24;
    }
};
int main()
{
    system("title Umar Farooq-088-Lab2-1B");
    Time t1;
    t1.getTime();
    cout << endl;
    t1.displaytime();
    cout << endl;
    Time t2;
    t2.getTime();
    cout << endl;
    t2.displaytime();
    cout << endl;
    Time add;
    add.addTime(t1, t2);
    add.displaytime();
    cout << endl;
    cout << "Number of Days: " << add.numberOfDays() << endl;
    system("pause");
}

```

#### Output:



```

Umar Farooq-088-Lab2-1B
Enter hour: 24
Enter Minute: 53
Enter Seconds: 63

Time: 24:53:63

Enter hour: 41
Enter Minute: 24
Enter Seconds: 67

Time: 41:24:67

After adding both times: Time: 65:77:130

Number of Days: 2
Press any key to continue . . .

```



In case of output snippet please make sure output snippet contains student name and id. AliAhmed\_123\_Lab02\_T1.exe

**Task 02:** C++ program to create a class to read and add two distance, Distance should be input in the form of feet and inch and check if inch is greater than 12 and increment it into feet and display remaining inch and display using the given function.

Write functions:

```
void getDist ();  
void showDist ();  
Distance addDist( Distance );  
Distance subDist( Distance );
```

**Solution:**

Enter your code here...

```
#include <iostream>  
#include <cstdlib>  
using namespace std;  
class Distance {  
private:  
    int feet;  
    int inches;  
public:  
    void setdata(int F, int I)  
    {  
        feet = F;  
        inches = I;  
    }  
    void getDist()  
    {  
        cout << "Enter Feet: ";  
        cin >> feet;  
        cout << "Enter Inches: ";  
        cin >> inches;  
    }  
    void showDist()  
    {  
        cout << feet << "ft " << inches << "in" << endl;  
    }  
    void addDist (Distance d1, Distance d2)  
    {  
        feet = d1.feet + d2.feet;  
        inches = d1.inches + d2.inches;
```

```

        if (inches >= 12)
        {
            feet += inches / 12;
            inches = inches % 12;
        }
    }

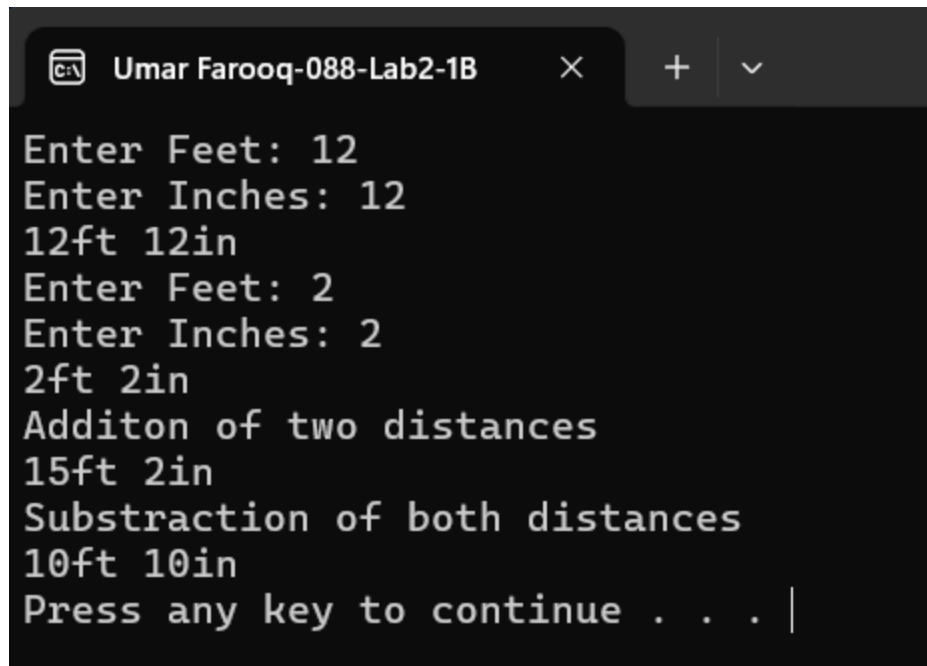
    void subsDist(Distance d1, Distance d2)
    {
        feet = d1.feet - d2.feet;
        inches = d1.inches - d2.inches;
    }
};
int main()
{
    Distance d1;
    d1.getDist();
    d1.showDist();
    Distance d2;
    d2.getDist();
    d2.showDist();
    Distance op;
    cout << "Additon of two distances" << endl;
    op.addDist(d1, d2);
    op.showDist();

    cout << "Substraction of both distances" << endl;
    op.subsDist(d1, d2);
    op.showDist();
}

```

**Output:**

Your Output here...



```
Umar Farooq-088-Lab2-1B
Enter Feet: 12
Enter Inches: 12
12ft 12in
Enter Feet: 2
Enter Inches: 2
2ft 2in
Additon of two distances
15ft 2in
Substraction of both distances
10ft 10in
Press any key to continue . . . |
```

In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab02_T2.exe`