

শিক্ষা নিয়ে গড়বো দেশ

তথ্য-প্রযুক্তির বাংলাদেশ

**Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh**



# LAB REPORT-03

---

**COURSE NO.- PROG 112**  
**COURSE TITLE- OBJECT ORIENTED**  
**PROGRAMMING SESSIONAL**

## **SUBMITTED BY**

Mehrin Farzana

ID: 2101013

Department of IRE

Session :2021-2022

Bangabandhu Sheikh Mujibur Rahman Digital  
University, Bangladesh

## **SUBMITTED TO**

Md.Toukir Ahmed

Lecturer

Department of IRE

Bangabandhu Sheikh Mujibur Rahman Digital  
University, Bangladesh

---

**Date of Submission: 31 July, 2023**

## Problem No.: 01

**Problem Name:** Classes, Public and Private access modifiers in C++.

### Code:

```
#include<iostream>
using namespace std;
class Employee{
    private:
        int a,b,c;
    public:
        int d,e;
        void setData(int a1, int b1, int c1);
        void getData(){
            cout<<"a = "<<a<<endl;
            cout<<"b = "<<b<<endl;
            cout<<"c = "<<c<<endl;
            cout<<"d = "<<d<<endl;
            cout<<"e = "<<e<<endl;
        }
};
void Employee::setData(int a1, int b1, int c1){
    a=a1;
    b=b1;
    c=c1;
}
int main(){
    Employee harry;
    harry.d=67;
    harry.e=98;
    harry.setData(1,2,4);
    harry.getData();
    return 0;
}
```

## Output:



```
"C:\Users\BAB AL SAFA\OneD" × + v
a = 1
b = 2
c = 4
d = 67
e = 98

Process returned 0 (0x0)   execution time : 0.055 s
Press any key to continue.
|
```

Fig 1.1: Output on console.

## Explanation:

The code defines a class named Employee with private members a,b,c and public members d,e. The class has two member functions setData() and getData(). setData() is used to set the values of private members a,b,c and getData() is used to display the values of all members.

The line `#include<iostream>` is used to include the input/output stream library in the program. The line `using namespace std;` is used to avoid writing `std::` before every standard library function call.

The main function creates an object of the Employee class named harry. The values of public members d and e are set to 67 and 98 respectively using the dot operator. The function setData() is called with arguments 1,2,4 which sets the values of private members a,b,c. Finally, getData() function is called which displays all the member variables.

## **Problem No.: 02**

**Problem Name:** Protected Access Modifier in C++.

### **Code:**

```
#include<iostream>
using namespace std;
class Base{
    protected:
        int a;
    public:
        int b;
};
class Derived: protected Base{

};

int main(){
    Base b;
    Derived d;
    cout<<b.a; //Error
    return 0;
}
```

## Output:

File	L...	Message
		=== Build: Debug in tst cpp1 (compiler: GNU GCC Compiler) ===
C:\Users...		In function 'int main()':
C:\Users...	16	error: 'int Base::a' is protected within this context
C:\Users...	5	note: declared protected here
C:\Users...	15	warning: unused variable 'd' [-Wunused-variable]
		=== Build failed: 1 error(s), 1 warning(s) (0 minute(s), 0 ...

Fig 2.1: Error on panel.

## Explanation:

The code defines two classes named Base and Derived. The class Base has a protected member variable a and a public member variable b. The class Derived is derived from the Base class using protected inheritance.

Protected inheritance is similar to private inheritance but with one difference. In private inheritance, all the public and protected members of the base class become private members of the derived class. In protected inheritance, all the public and protected members of the base class become protected members of the derived class.

The main function creates an object of the Base class named b and an object of the Derived class named d. The line `cout<<b.a;` tries to access the protected member variable a of the Base class which is not allowed outside the class.

### **Problem No.: 03**

**Problem Name:** Encapsulation In C++.

#### **Code:**

```
#include<iostream>
using namespace std;
class Company{
    private:
        int salary;

    public:
        void setSalary(int fulltime, int overtime){
            salary = fulltime+overtime;
        }
        int getSalary(){
            return salary;
        }
};

int main(){
    Company obj;
    obj.setSalary(30000,6000);
    cout<<"Total salary: "<<obj.getSalary()<<endl;
    return 0;
}
```

## Output:

A screenshot of a Windows command prompt window. The title bar shows the path "C:\Users\BAB AL SAFA\OneD" with a close button. The command prompt displays the following text: "Total salary: 36000", "Process returned 0 (0x0) execution time : 3.992 s", and "Press any key to continue." The text is in a monospaced font on a black background.

Fig 3.1: Output on console.

## Explanation:

The code defines a class named `Company` with a private member variable `salary` and two member functions `setSalary()` and `getSalary()`.

The function `setSalary()` takes two integer arguments `fulltime` and `overtime` and sets the value of `salary` to the sum of `fulltime` and `overtime`. The function `getSalary()` returns the value of `salary`.

The main function creates an object of the `Company` class named `obj`. The function `setSalary()` is called with arguments `30000` and `6000` which sets the value of `salary` to `36000`. Finally, the function `getSalary()` is called which returns the value of `salary` which is then printed to the console using `cout`.

## **Problem No.: 04**

**Problem Name:** Encapsulation In C++.

### **Code:**

```
#include<iostream>
using namespace std;
class Ball{
    private:
        float area;

    public:
        void setArea(int radius){
            area = 3.14*radius*radius;
        }
        float getArea(){
            return area;
        }
};

int main(){
    Ball obj;
    obj.setArea(6);
    cout<<"Area of ball is: "<<obj.getArea()<<endl;
    return 0;
}
```



## Output:

A screenshot of a Windows command prompt window. The title bar shows the path "C:\Users\BAB AL SAFA\OneD" with a close button. The command prompt displays the following text: "Area of ball is: 113.04", "Process returned 0 (0x0) execution time : 3.455 s", and "Press any key to continue." The text is in a monospaced font on a black background.

Fig 4.1: Output on console.

## Explanation:

The code defines a class named Ball with a private member variable area and two member functions setArea() and getArea().

The function setArea() takes an integer argument radius and sets the value of area to the area of the ball with that radius. The function getArea() returns the value of area.

The main function creates an object of the Ball class named obj. The function setArea() is called with argument 6 which sets the value of area to  $3.1466 = 113.04$ . Finally, the function getArea() is called which returns the value of area which is then printed to the console using cout.

## Problem No.: 05

**Problem Name:** Abstract Base Class & Pure Virtual Functions in C++ .

### Code:

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

class CWH{
protected:
    string title;
    float rating;

public:
    CWH(string s, float r){
        title = s;
        rating = r;
    }
    virtual void display()=0;
};

class CWHVideo: public CWH{
    float videoLength;
public:
    CWHVideo(string s, float r, float vl): CWH(s,r){
        videoLength = vl;
    }
    void display(){
        cout<<"This is an amazing video with title "<<title<<endl;
        cout<<"Rating: "<<rating<<" out of 5 stars"<<endl;
        cout<<"Length of this video is "<<videoLength<<" minutes"<<endl;
    }
};

class CWHText: public CWH{
    float words;
public:
    CWHText(string s, float r, float wc): CWH(s,r){
        words = wc;
    }
    void display(){
        cout<<"This is an amazing text tutorial with title "<<title<<endl;
        cout<<"Ratings of this text tutorial: "<<rating<<" out of 5 stars"<<endl;
    }
};
```

```

        cout<<"No of words in this text tutoria is "<<words<<" minutes"<<endl;
    }
};

```

```

int main(){
    string title;
    float rating, vlen;
    int words;

    title = "Django tutorial";
    vlen=4.56;
    rating=4.89;
    CWHVideo djVideo(title, rating, vlen);

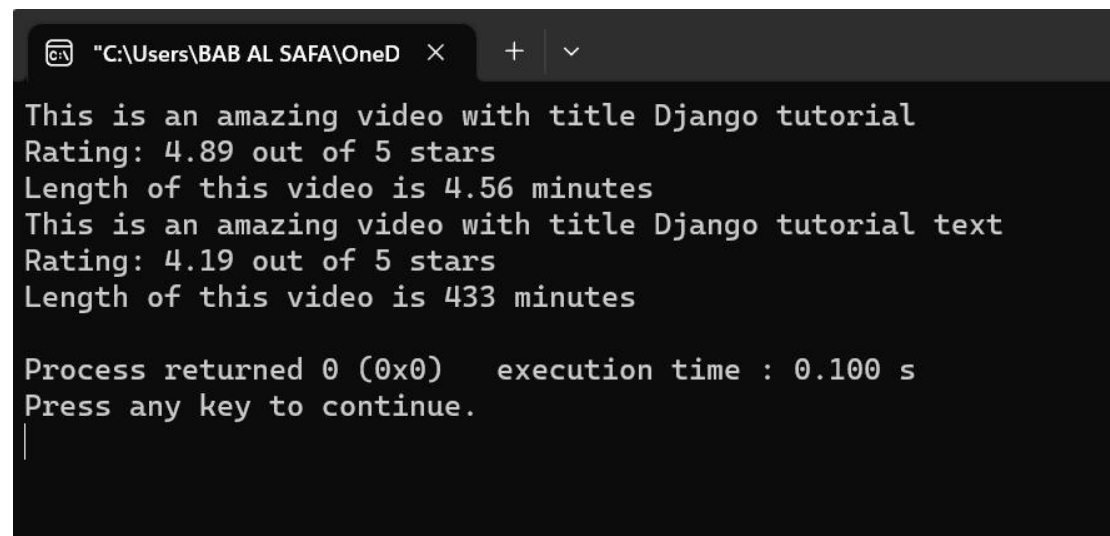
    title = "Django tutorial text";
    words=433;
    rating=4.19;
    CWHVideo djText(title, rating, words);

    CWH *tuts[2];
    tuts[0]=&djVideo;
    tuts[1]=&djText;

    tuts[0]->display();
    tuts[1]->display();
    return 0;
}

```

## Output:



```
"C:\Users\BAB AL SAFA\OneD"
This is an amazing video with title Django tutorial
Rating: 4.89 out of 5 stars
Length of this video is 4.56 minutes
This is an amazing video with title Django tutorial text
Rating: 4.19 out of 5 stars
Length of this video is 433 minutes

Process returned 0 (0x0)   execution time : 0.100 s
Press any key to continue.
|
```

Fig 5.1: Output on console

## Explanation:

This is a C++ program that demonstrates the use of inheritance and polymorphism. It creates two classes, CWHVideo and CWHText, both of which inherit from the CWH class. The CWH class is an abstract class that has a pure virtual function called display(). The CWHVideo and CWHText classes implement this function in their own way. The program creates two objects of these classes and stores them in an array of pointers to the base class. It then calls the display() function on each object using the pointer to the base class.

Here's a brief explanation of the code:

- The program starts by including the necessary header files.
- A class called CWH is defined with two protected data members: a string called title and a float called rating. It also has a public constructor that takes two arguments: a string called s and a float called r. This constructor initializes the data members with the values passed as arguments.
- The CWH class also has a pure virtual function called display().
- Two classes called CWHVideo and CWHText are defined that inherit from the CWH class.
- The CWHVideo class has an additional data member called videoLength, which is a float.
- The CWHText class has an additional data member called words, which is an int.
- Both classes implement the display() function in their own way.
- In the main function, two objects of these classes are created and stored in an array of pointers to the base class.
- The program then calls the display() function on each object using the pointer to the base class.