



# **OBJECT ORIENTED PROGRAMMING**

## **Lab Report #04**

**Section “C”**

Submitted To:  
**Sir Muhammad Husnain**

Submitted by:

**Abdul Ahad**  
**22-CS-071**

**DEPARTMENT OF CS**  
**HITEC UNIVERSITY, TAXILA**

## Example 1:

### Code:

```
#include <iostream>

using namespace std;

const double pi = 3.14159;

class Cylinder {
public:
    double radius;
    double height;

    double calculateArea() {
        return ((2 * pi * radius * height) + (2 * pi * radius * radius));
    }

    double calculateVolume() {
        return (pi * radius * radius * height);
    }
};

int main() {
```

```
Cylinder can;

can.radius = 4;

can.height = 10;

cout << "Area of can = " << can.calculateArea() << endl;

cout << "Volume of can = " << can.calculateVolume() << endl;

return 0;

}
```

## Description:

This is a code that creates a class “Cylinder”, which can store the basic properties of a cylinder such as its height and radius. It also utilizes methods to calculate the area and volume of the cylinder.

## Results:

```
Area of can = 351.858
Volume of can = 502.654

-----
Process exited after 0.05559 seconds with return value 0
Press any key to continue . . .
```

## Example 2

## Code:

```
#include <iostream>

using namespace std;

const double pi = 3.14159;

class Cylinder {

private:

double radius;

double height;

public:

void initData(double rad, double hgt) {

radius = rad;

height = hgt;

}

double calculateArea() {

return ((2 * pi * radius * height) + (2 * pi * radius * radius));

}
```

```
double calculateVolume() {  
    return (pi * radius * radius * height);  
}  
};  
  
int main() {  
  
    Cylinder can;  
  
    can.radius = 4;  
    can.height = 10;  
  
    cout << "Area of can = " << can.calculateArea() << endl;  
    cout << "Volume of can = " << can.calculateVolume() << endl;  
    return 0;  
}
```

## Description:

This is the same code as in example #1. However, the data members of the “Cylinder” class in this code are set to the access specifier : Private, and hence cannot be accessed outside the class. For this purpose, I have used functions to initialize the data members with their values, which are then passed onto the Area and Volume calculating methods.

## Results:

```
Area of can = 351.858
Volume of can = 502.654

-----
Process exited after 0.04712 seconds with return value 0
Press any key to continue . . .
```

## Example 3:

## Code:

```
#include <iostream>

using namespace std;

class MyClass {

public:

    int myNum;

    string myString;

};

int main() {

    MyClass myObj;

    cout << "Enter a number: ";

    cin >> myObj.myNum;

    cout << "Enter a word: ";

    cin >> myObj.myString;

    cout << myObj.myNum << "\n";

    cout << myObj.myString;

    return 0;

}
```

## Description:

This is a code that creates a class “MyClass”, which has the data members myNum and myString, used to store an integer and a string respectively. This code is an example of a class with public data members and how these data members can be accessed even outside of the class.

## Results:

```
Enter a number: 13
Enter a word: abcde
13
abcde
-----
Process exited after 5.027 seconds with return value 0
Press any key to continue . . .
```

## Example 4:



## Code:

```
#include <iostream>

using namespace std;

class Employee {
private:
    int salary;

public:
    void setSalary(int s) {
        salary = s;
    }
    int getSalary() {
        return salary;
    }
};

int main() {
    Employee myObj;

    int a;
```

```
cout << "Enter your salary: ";  
  
cin >> a;  
  
myObj.setSalary(a);  
  
cout << "Your salary is: $" << myObj.getSalary();  
  
return 0;  
  
}
```

## Description:

This code continues the objective of examples 1 and 2, displaying the use of private data members for abstraction and protection of data. However, this code is an example that displays the process of inputting data from a user and then using class methods to initialize the data members of the class.

## Results:

```
Enter your salary: 1300  
Your salary is: $1300  
-----  
Process exited after 1.664 seconds with return value 0  
Press any key to continue . . .
```

## Example 5:

## Code:

```
#include<iostream>

using namespace std;

void print(int i, int j, int rr){

    int a=i;

    int b=j;

    int r=rr;

    int c=(a+b)/2;

    cout<<"\n\nRegistration number of : "<<r<<"\nAverage marks
are : "<<c<<endl;

}

int main(){

    int m1;

    int m2;

    int reg;

    for(int i = 0; i < 2; i++){
```

```
cout<<"Enter your Registration Number: ";  
  
cin>>reg;  
  
cout<<"\nEnter marks in Mathematics: ";  
  
cin>>m1;  
  
cout<<"\nEnter marks in Computer Science: ";  
  
cin>>m2;  
  
print(m1,m2,reg);  
  
cout << "\n\n";  
  
}
```

**Description:**

This isn't an example of classes. However, it is an example of functions which are deeply related to the concept of classes and consistently used in them. This code features a functions that takes input the registration and marks of a user and outputs their registration number and their average marks. A loop is also used to allow for the data input of two users.

## Results:

```
Enter your Registration Number: 13
Enter marks in Mathematics: 50
Enter marks in Computer Science: 50

Registration number of :13
Average marks are : 50

Enter your Registration Number: 71
Enter marks in Mathematics: 81
Enter marks in Computer Science: 94

Registration number of :71
Average marks are : 87

-----
Process exited after 12.31 seconds with return value 0
Press any key to continue . . .
```