National University of Computer & Emerging Sciences, Karachi
Computer Science Department
Fall 2021, Lab Manual - 03

Course Code: CL-217	Course : Object Oriented Programming Lab

Lab # 03

Contents:

- Classes
- Objects
- Public and Private Keywords
- Structures VS Classes
- Transformation from Procedural to Object Oriented Programming
- Example Programs
- Exercise

Classes

A class is a programmer-defined data type that describes what an object of the class will look like when it is created. It consists of a set of variables and a set of functions.

We can think of class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows etc. Based on these descriptions we build the house. House is the object.

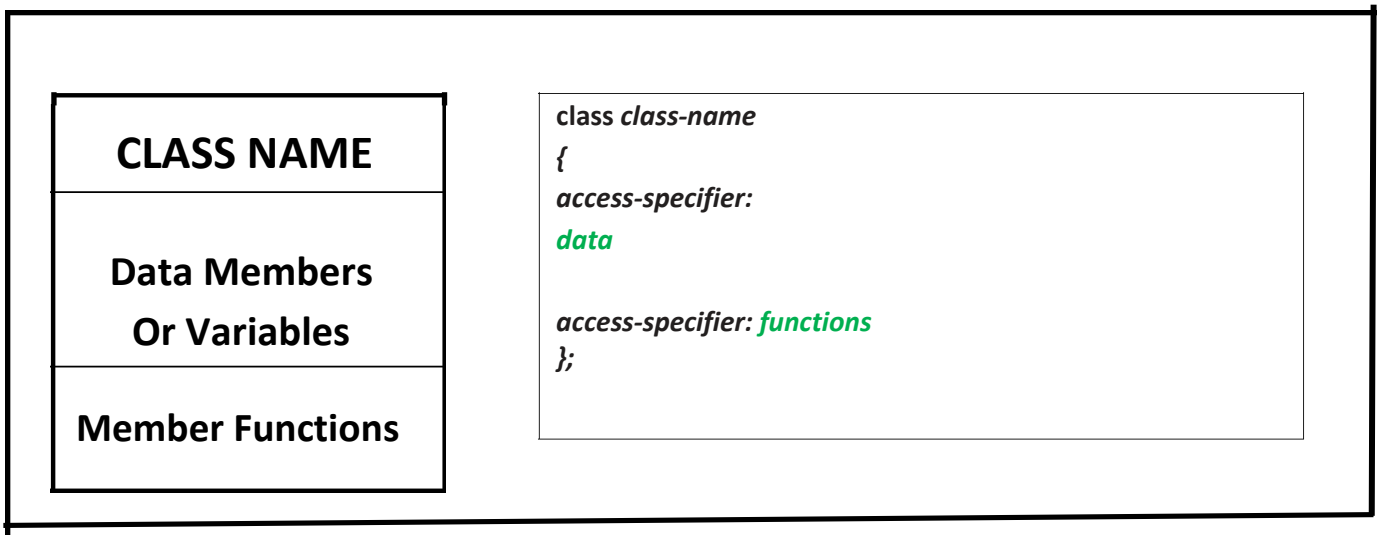
As, many houses can be made from the same description, we can create many objects from a class.

Classes are created using the keyword **class**. A class declaration defines a new type that links code and data. This new type is then used to declare objects of that class.

- A Class is a user defined data-type which has data members and member functions.
- Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions defines the properties and behavior of the objects in a Class.

A class member can be defined as public, private or protected. By default members would be assumed as private.

In the UML, a class icon can be subdivided into compartments. The top compartment is for the name of the class, the second is for the variables of the class, and the third is for the methods of the class.



CLASS NAME

By convention, the name of a user-defined class begins with a capital letter and, for readability, each subsequent word in the class name begins with a capital letter.

DATA MEMBERS

Consider the attributes of some real world objects:

RADIO – station setting, volume setting.
CAR – speedometer readings, amount of gas in its tank and what gear it is in.

These attributes form the data in our program. The values that these attributes take (the blue color of the petals, for example) form the state of the object.

MEMBER FUNCTIONS

Consider the operations of some real world objects:

RADIO – setting its station and volume (invoked by the person adjusting the radio's controls)
CAR – accelerating (invoked by the driver), decelerating, turning and shifting gears. These operations form the functions in program. Member functions define the class's behaviors.

Objects

In C++, when we define a variable of a class, we call it **instantiating** the class. The variable itself is called an **instance** of the class. A variable of a class type is also called an **object**. Instantiating a variable allocates memory for the object.

Syntax to Define Object in C++

```
className objectVariableName;
```

RADIO r;

CAR c;

Accessing Public Data Members

The public data members of objects of a class can be accessed using the direct member access operator (.).

However the private data members are not allowed to be accessed directly by the object. Accessing a data member depends solely on the access control of that data member.

Accessing Private Data Members

To access, use and initialize the private data member you need to create getter and setter functions, to get and set the value of the data member.

The setter function will set the value passed as argument to the private data member, and the getter function will return the value of the private data member to be used. Both getter and setter function must be defined public.

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

class Student
{
    private:    // private data member
    int rollno;

    public:
    // public function to get value of rollno - getter
    int getRollno()
    {
        return rollno;
    }
    // public function to set value for rollno - setter
    void setRollno(int i)
    {
        rollno=i;
    }
};

int main()
{
    Student A;
    A.rollno=1;  //Compile time error
    cout<< A.rollno; //Compile time error

    A.setRollno(1);  //Rollno initialized to 1
    cout<< A.getRollno(); //Output will be 1
}
```

Member Functions in Classes

There are 2 ways to define a member function:

- Inside class definition
- Outside class definition

1. Inside class definition

With an inline function, the compiler tries to expand the code in the body of the function in place of a call to the function.
Note that all the member functions defined inside the class definition are by default **inline**, but you can also make any non-class function inline by using keyword inline with them. Inline functions are actual functions, which are copied everywhere during compilation, like pre-processor macro, so the overhead of function calling is reduced.

2. Outside class definition

To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

C++ > oop.cpp > main()

```cpp
#include <iostream>
using namespace std;
class Student
{
    public:
    string StudentName;
    int id;

    // printname is not defined inside class definition
    void printname();

    // printid is defined inside class definition
    void printid()
    {
        cout << " Student id is: " << id;
    }
};

// Definition of printname using scope resolution operator ::
void Student::printname()
{
    cout << " Student name is: " << StudentName;
}
int main() {

    Student obj1;
    obj1. StudentName = "xyz";
    obj1.id=15;

    // call printname()
    obj1.printname();
    cout << endl;

    // call printid()
    obj1.printid();
    return 0;
}
```

```
C:\Users\Administrator\Documents\C++>cd "c:\Users\Administrator\Documents\C++\" && g++ oop.cpp -o oop && "c:\Users\Administrator\Documents\C++\"
 Student name is: xyz
 Student id is: 15
```

Structures VS Classes

By default, all structure fields are public, or available to functions (like the main() function) that are outside the structure. Conversely, all class fields are private. That means they are not available for use outside the class. When you create a class, you can declare some fields to be private and some to be public. For example, in the real world, you might want your name to be public knowledge but your Social Security number, salary, or age to be private.

TRANSFORMATION FROM PROCEDURAL TO OBJECT ORIENTED PROGRAMMING

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

double calculateBMI(double w, double h)
{
 return w/(h*h)*703;
}

string findStatus(double bmi)
{
 string status;
if(bmi < 18.5)
  status = "underweight";
else if(bmi < 25.0)
  status = "normal";// so on.
return status;
}

int main()
{
    double bmi, weight, height;
    string status;
    cout<<"Enter weight in Pounds ";
    cin>>weight;
    cout<<"Enther height in Inches ";
    cin>>height;
    bmi=calculateBMI(weight,height);
    cout<<"Your BMI is "<<bmi<<" Your status is "<<findStatus(bmi);

}
```

Procedural Approach

```
#include<iostream>
using namespace std;
class BMI
{
    double weight, height,bmi;
    string status;
    public:
        void getInput() {
            cout<<"Enter weight in Pounds ";
            cin>>weight;
            cout<<"Enther height in Inches ";
            cin>>height;   }
        double calculateBMI() {
             return weight / (height*height)*703; }
         string findStatus() {
            if(bmi < 18.5)
                status = "underweight";
            else if(bmi < 25.0)
                status = "normal";// so on.
            return status; }
        void printStatus()  {
            bmi = calculateBMI();
            cout<< "You BMI is "<< bmi<< "your status is " << findStatus(); }
};

int main()
{
    BMI bmi;
    bmi.getInput();
    bmi.printStatus();
}
```

Object Oriented Approach

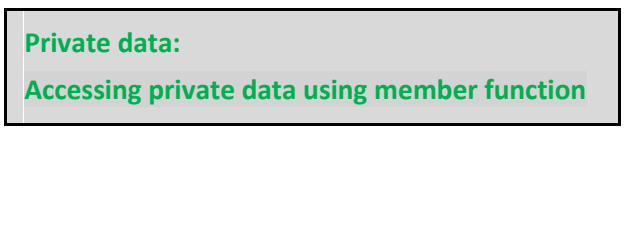
EXAMPLE PROGRAM

```
#include<iostream>
using namespace std;
class Account
{
private:
        double balance; // Account balance public:
//Public interface:
        string name; // Account holder long accountNumber;
        // Account number void setDetails(double bal)
        {
                balance = bal;
        }
        double getDetails()
        {
                return balance;
        }
        void displayDetails()
        {
                cout<<"Details are: "<<endl;
           cout<<"Account Holder:
           "<<name<<endl;
                cout<<"Account Number:
                "<<
           accountNumber <<endl; cout<<"Account
            Balance: "<<getDetails()<<endl;
        }
};

int main(){ double accBal; Account
currentAccount;
currentAccount.getDetails();

cout<<"Please enter the details"<<endl;
cout<<"Enter Name:"<<endl; getline(cin,
currentAccount.name); cout<<"Enter
Account Number:"<<endl;
cin>>currentAccount.accountNumber;

cout<<"Enter Account
```

Set and get functions to manipulate private data member

Publicly available data: Assigning values from

```
Balance:"<<endl; cin>>accBal;
currentAccount.setDetails(accBal);
cout<<endl;
currentAccount.displayDetails();
return 0;
        }
```



LAB TASKS

Question # 01:

Define a class to represent a student. Include the following members

Data Members:

- Name of the student
- ID number
- Department name
- Six Subjects
- Marks obtained
- Total marks

Member Functions:

- Write setter and getter property to set all attributes
- To calculate percentage and grade
- To display name and ID number, percentage and grade

Question # 02:

A bank wants a simple application module to manage the accounts of its customers. For every new customer, the app must let us fill in the details including his Name, Age, NIC#, Address, Opening Balance, Current Balance, Contact# & PIN. These details may be modified later except for the PIN.

At any given time, the customer can check his balance.

Tax must be calculated (Tax is 0.15% of the current balance for customers aged 60 or above and 0.25% for all other customers).

Question # 03:

Create a class called Bank Employee that includes three pieces of information as data members—a first name (type char* (DMA)), a last name (type string) and a monthly salary (type int). Your class should have a setter function that initializes the three data members. Provide a getter function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create five Employee objects and display each object's yearly salary. Then give each Employee a 20 percent raise and display each Employee's yearly salary again. Identify and add any other related functions to achieve the said goal.

Home Assignment

Question # 04:

Define a class for a type called **CounterType**. An object of this type is used to count things, so it records a count that is a nonnegative whole number. Include a mutator function that sets the counter to a count given as an argument. Include member functions to increase the count by one and to decrease the count by one. Also, include a member function that returns the current count value and one that outputs the count. Embed your class definition in a test program.

Question # 05:

You are a programmer for the **Standard Charted Bank** assigned to develop a class that models the basic workings of a bank **account**. The class should perform the following tasks:

- Save the account balance.
- Save the number of transactions performed on the account.
- Allow deposits to be made to the account.
- Allow with drawls to be taken from the account.
- Calculate interest for the period.
- Report the current account balance at any time.
- Report the current number of transactions at any time.

Menu

1. Display the account balance
2. Display the number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Add interest for this period
7. Exit the program