# Department of Electrical Engineering

**CS212**

**Object Oriented Programming**



# Lab 4: Introduction to Classes

**Class**: BEE - 12C

**Date**: October 15th, 2021

**Time**: Monday (1400 – 1700)

**Name**: Muhammad Umer

**CMS ID**: 345834

**Objectives**

The objective of this lab is an introduction to the concept of classes & objects in C++. Moreover, we will also explore access modifiers in today’s lab.

**Tools**

* Microsoft Visual Studio 2013

The mechanism that allows you to combine data and the function in a single unit is called a class. Once a class is defined, you can declare variables of that type. A class variable is called object or instance. In other words, a class would be the data type, and an object would be the variable. Classes are generally declared using the keyword class, with the following format:

class class\_name

{

   private:

      members1;

   protected:

      members2;

   public:

      members3;

};

Where class\_name is a valid identifier for the class. The body of the declaration can contain members, that can be either data or function declarations, The members of a class are classified into three categories: private, public, and protected. Private, protected, and public are reserved words and are called member access specifiers. These specifiers modify the access rights that the members following them acquire.

**private members** of a class are accessible only from within other members of the same class. You cannot access it outside of the class.  
**protected members** are accessible from members of their same class and also from members of their derived classes.   
Finally, **public members** are accessible from anywhere where the object is visible.

By default, all members of a class declared with the class keyword have private access for all its members. Therefore, any member that is declared before one other class specifier automatically has private access.

Here is a complete example:

class student

{

  private :

    int rollno;

    float marks;

  public:

    void getdata()

    {

       cout<<"Enter Roll Number : ";

       cin>>rollno;

       cout<<"Enter Marks : ";

       cin>>marks;

    }

    void displaydata()

    {

       cout<<"Roll number : "<<rollno<<"\nMarks : "<<marks;

    }

};

**Object Declaration**

Once a class is defined, you can declare objects of that type. The syntax for declaring a object is the same as that for declaring any other variable. The following statements declare two objects of type student:

student st1, st2;

## Accessing Class Members

Once an object of a class is declared, it can access the public members of the class.

st1.getdata();

## Defining Member function of class

You can define Functions inside the class as shown in above example. Member functions defined inside a class this way are created as inline functions by default. It is also possible to declare a function within a class but define it elsewhere. Functions defined outside the class are not normally inline.  
When we define a function outside the class we cannot reference them (directly) outside of the class. In order to reference these, we use the scope resolution operator, :: (double colon). In this example, we are defining function getdata outside the class

void student :: getdata()

{

     cout<<"Enter Roll Number : ";

     cin>>rollno;

     cout<<"Enter Marks : ";

     cin>>marks;

}

The following program demostrates the general feature of classes. Member function initdata() is defined inside the class. Member funcitons getdata() and showdata() defined outside the class.

class student //specify a class

{

  private :

    int rollno; //class data members

    float marks;

  public:

    void initdata(int r, int m)

    {

       rollno=r;

       marks=m;

    }

    void getdata(); //member function to get data from user

    void showdata();// member function to show data

};

void student :: getdata()

{

    cout<<"Enter Roll Number : ";

    cin>>rollno;

    cout<<"Enter Marks : ";

    cin>>marks;

}

void student :: showdata()   // Display the data

{

    cout<<"Roll number : "<<rollno<<"\nMarks : "<<marks;

}

int main()

{

    student st1, st2; //define two objects of class student

    st1.initdata(5,78); //call member function to initialize

    st1.showdata();

    st2.getdata(); //call member function to input data

    st2.showdata(); //call member function to display data

    return 0;

}

**Lab Tasks**

* **Task 1**

**Q1.** Create an ATM system through use of Classes.

#include <iostream>

#include <stdlib.h>

#include <string.h>

using namespace std;

class Bank {

private:

    string name;

    int accnumber;

    char type[10];

    int amount = 0;

    int tot = 0;

public:

    void setvalue()

    {

        cout << "Enter name\n";

        cin.ignore();

        getline(cin, name);

        cout << "Enter Account number\n";

        cin >> accnumber;

        cout << "Enter Account type\n";

        cin >> type;

        cout << "Enter Balance\n";

        cin >> tot;

    }

    void showdata()

    {

        cout << "Name:" << name << endl;

        cout << "Account No: " << accnumber << endl;

        cout << "Account type: " << type << endl;

        cout << "Balance: " << tot << endl;

    }

    void deposit()

    {

        cout << "\nEnter amount to be Deposited\n";

        cin >> amount;

    }

    void showbal()

    {

        tot = tot + amount;

        cout << "\nTotal balance is: " << tot;

    }

    void withdrawl()

    {

        int a, avai\_balance;

        cout << "Enter amount to withdraw\n";

        cin >> a;

        avai\_balance = tot - a;

        cout << "Available Balance is" << avai\_balance;

    }

};

int main()

{

    Bank b;

    int choice;

    while (1) {

        cout << "\t\t WELCOME!\n";

        cout << "Enter Your Choice\n";

        cout << "\t1. Enter name, Account "

             << "number, Account type\n";

        cout << "\t2. Balance Enquiry\n";

        cout << "\t3. Deposit Money\n";

        cout << "\t4. Show Total balance\n";

        cout << "\t5. Withdraw Money\n";

        cout << "\t6. Cancel\n";

        cin >> choice;

        switch (choice) {

        case 1:

            b.setvalue();

            break;

        case 2:

            b.showdata();

            break;

        case 3:

            b.deposit();

            break;

        case 4:

            b.showbal();

            break;

        case 5:

            b.withdrawl();

            break;

        case 6:

            exit(1);

            break;

        default:

            cout << "\nInvalid choice\n";

        }

    }

}

**Terminal Output**

                 WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

1

Enter name

Muhammad Umer

Enter Account number

345834

Enter Account type

Current

Enter Balance

65322

                 WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

2

Name:Muhammad Umer

Account No: 345834

Account type: Current

Balance: 65322

                 WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

3

Enter amount to be Deposited

543

                 WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

4

Total balance is: 65865          WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

5

Enter amount to withdraw

6785

Available Balance is59080                WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

4

Total balance is: 66408          WELCOME!

Enter Your Choice

        1. Enter name, Account number, Account type

        2. Balance Enquiry

        3. Deposit Money

        4. Show Total balance

        5. Withdraw Money

        6. Cancel

6

* **Task 2**

**Q2.** Define a class named Movie. Include private fields for the title, year, and name of the director. Include three public functions with the prototypes void Movie::setTitle(string);, void Movie::setYear(int);, and void setDirector(string);. Include another function that displays all the information about a Movie. Write a main() function that declares a movie object named myFavoriteMovie. Set and display the object’s fields. Save the file as Movie.cpp.

#include <iostream>

#include <string>

using namespace std;

// class declaration

class Movie {

 private:

  string movieTitle;

  int movieYear;

  string directorName;

 public:

  void setTitle(string title);

  void setYear(int year);

  void setDirector(string director);

  void displayInfo();

};

// class Implementation

void Movie::setTitle(string title) {

  movieTitle = title;

}

void Movie::setYear(int year) {

  movieYear = year;

}

void Movie::setDirector(string director) {

  directorName = director;

}

void Movie::displayInfo() {

  cout << endl;

  cout << "Movie Title: " << movieTitle << endl;

  cout << "Year: " << movieYear << endl;

  cout << "Director: " << directorName << endl;

}

// main function

int main() {

  Movie myFavoriteMovie;

  string title, name;

  int year;

  cout << "Enter movie title: " << endl;

  getline(cin, title);

  myFavoriteMovie.setTitle(title);

  cout << "Enter Director's Name: " << endl;

  getline(cin, name);

  myFavoriteMovie.setDirector(name);

  cout << "Enter movie Year: " << endl;

  cin >> year;

  myFavoriteMovie.setYear(year);

  // display all the data

  myFavoriteMovie.displayInfo();

  system("PAUSE");

  return 0;

}

**Terminal Output**

Enter movie title:

The Shawshank Redemption

Enter Director's Name:

Frank Darabont

Enter movie Year:

1994

Movie Title: The Shawshank Redemption

Year: 1994

Director: Frank Darabont

* **Task 3**

**Q3.** Computer games often contain different characters or creatures. For example, you might design a game in which alien beings possess specific characteristics such as color, number of eyes, and number of legs. Design a character for a game, creating a class to hold at least three attributes for each character. Include methods to set each of the character’s attributes. Then write an application(i.e main function) in which you create at least two characters (i.e objects), each of which has a random age up to 100, a random number of eyes up to 10, and a random number of legs, up to 12. In turn, pass each character to a display method that displays the character’s attributes. Save the program as MyCharacters.cpp.

#include <iostream>

#include <string>

#include <ctime>

#include <cstdlib>

using namespace std;

// declaration

class Alien {

 private:

  string color;

  int legs;

  int eyes;

  int age;

 public:

  string getColor() { return color; }

  void setColor(string new\_color) { color = new\_color; }

  int getLegs() { return legs; }

  void setLegs(int new\_legs) { legs = rand() % new\_legs; }

  int getEyes() { return eyes; }

  void setEyes(int new\_eyes) { eyes = rand() % new\_eyes; }

  int getAge() { return age; }

  void setAge(int new\_age) { age = rand() % new\_age; }

  void DisplayAlien() {

    cout << "Age: " << getAge() << endl;

    cout << "Legs: " << getLegs() << endl;

    cout << "Eyes: " << getEyes() << endl;

    cout << "Color: " << getColor() << endl;

  }

};

// main

int main() {

  srand((unsigned)time(NULL));

  // first Alien

  cout << "Creating a new Alien 'first'" << endl;

  Alien first;

  cout << "Setting first Alien color." << endl;

  first.setColor("Green");

  cout << "Setting first Alien age." << endl;

  first.setAge(100);

  cout << "Setting first Alien legs." << endl;

  first.setLegs(12);

  cout << "Setting first Alien eyes." << endl;

  first.setEyes(10);

  cout << endl;

  // second Alien

  cout << "Creating a new Alien 'second'" << endl;

  Alien second;

  cout << "Setting second Alien color." << endl;

  second.setColor("Brown");

  cout << "Setting second Alien age." << endl;

  second.setAge(100);

  cout << "Setting second Alien legs." << endl;

  second.setLegs(12);

  cout << "Setting second Alien eyes." << endl;

  second.setEyes(10);

  cout << endl;

  cout << " First Alien information:" << endl;

  first.DisplayAlien();

  cout << endl;

  cout << " Second Alien information:" << endl;

  second.DisplayAlien();

}

**Terminal Output**

Creating a new Alien 'first'

Setting first Alien color.

Setting first Alien age.

Setting first Alien legs.

Setting first Alien eyes.

Creating a new Alien 'second'

Setting second Alien color.

Setting second Alien age.

Setting second Alien legs.

Setting second Alien eyes.

 First Alien information:

Age: 87

Legs: 6

Eyes: 0

Color: Green

 Second Alien information:

Age: 38

Legs: 5

Eyes: 7

Color: Brown