Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**3**
Student Name:	**Bhagyesh Subhash Manjrekar**
Roll No:	**07**

Title:

3.1 Write a C++ program to Create a class Student with two method getData() and printData(). getData() to get the value from the user and display the data in printData(). Create the two objects s1,s2 to declare and access the values from class StudentTest.

3.2 Write a C++ program for Basic bank Management System

Learning Objective:

- Students will be able to write C++ and java program for using classes and objects.

Learning Outcome:

- Ability to execute a simple C++ and Java program by accepting and displaying values using functions.
- Understanding the classes and objects concept in C++ and Java.

Course Outcome:

Understand object-oriented programming concepts and implement using C++ and Java.

Theory:

(1) Difference between procedural and object-oriented language.

=> Object-oriented programming and procedural programming both are used to develop the applications. Both of them are high-level programming languages; and it is also important to know the difference between them.

- **Procedural Language:**

It is defined as a programming language derived from the structure programming and based on calling procedures. The procedures are the functions, routines, or subroutines that consist of the computational steps required to be carried. As compared to object-oriented programming, procedural programming is less secure. Procedural programming follows a top-down approach during the designing of a program.

It gives importance to the concept of the function and divides the large programs into smaller parts or called as functions. Procedural programming is straightforward. Unlike object-oriented programming, there are no access modifiers introduced in procedural programming. The examples of procedural programming are BASIC, Pascal and C etc.

- **Object Oriented Programming language:**

In Object oriented programming, program is divided into small parts called objects**.** Object oriented programming follows bottom-up approach. Object oriented programming have access specifiers like private, public, protected etc. Object oriented programming provides data hiding so it is more secure.

Object oriented programming is based on real world. Overloading is possible in Object oriented programming. Object oriented programming is based on real world. The examples of Object oriented programming (oopm) programming are C++, Java, Python, C# etc.

(2) Application of object orientation programming in C++

=> OOPs stands for **Object-Oriented Programming**. It is about creating objects that contain both data and functions. Object-Oriented programming has several advantages over procedural languages. As OOP is faster and easier to execute it becomes more powerful than procedural languages like C++.

Application of OOPs are:

- User interface design such as windows, menu.
- Real Time Systems
- Simulation and Modelling

- Object oriented databases
- AI and Expert System
- Neural Networks and parallel programming
- Decision support and office automation systems etc.

With the help of inheritance, we can reuse the existing class to derive a new class such that the redundant code is eliminated and the use of existing class is extended. This saves time and cost of program. With the help of polymorphism, the same function or same operator can be used for different purposes. This helps to manage software complexity easily.

(3) **Brief introduction to C++**

=> **1.** C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a compiled language.

2. The name of C++ signifies the evolutionary nature of the changes from C. "++" is the C increment operator. C++ is one of the predominant languages for the development of all kind of technical and commercial software.

3. C++ introduces Object-Oriented Programming, not present in C. Like other things, C++ supports the four primary features of OOP: encapsulation, polymorphism, abstraction, and inheritance.

4. C++ is a cross-platform language that can be used to create high-performance applications. C++ was developed by Bjarne Stroustrup , as an extension to the C language. C++ gives programmers a high level of control over system resources and memory.

5. C++ is one of the world's most popular programming languages. ++ can be found in today's operating systems, Graphical User Interfaces, and embedded systems. C++ is portable and can be used to develop applications that can be adapted to multiple platforms.

6. C++ used in operating systems, game developments, IoT devices, Web browsers, financial tools, telecommunications, movie production etc.

PROGRAM 1: Write a C++ program to Create a class Student with two method getData() and printData(). getData() to get the value from the user and display the data in printData(). Create the two objects s1, s2 to declare and access the values from class StudentTest.

Algorithm:

Step 1: START

Step 2: Create class Student; with parameters which are needed as rollnum, name, branch, cgpa etc.

Step 3: Give two methods as getdata(), and printdata() in a class student.

Step 4: In main function create two objects s1,s2 to declare access values from class student.

Step 5: END

Program:

```
// To make student data
#include<iostream>
using namespace std;

class Student
{
        public:
        int rollnum;
        string name;
        string branch;
        float cgpa;

        int getdata()
```

```
    {
        cout<<"Enter your rollno :"<<endl;
    cin>>rollnum;
    cout<<"Enter your name :"<<endl;
    cin>>name;
    cout<<"Enter your branch :"<<endl;
    cin>>branch;
    cout<<"Enter the CGPA  :"<<endl;
    cin>>cgpa;
    }

    int printdata()
    {

    cout<<"Marksheet of student is as follows : "<<endl;
    cout<<"Roll number : "<<rollnum<<endl;
    cout<<"Name: "<<name<<endl;
    cout<<"Branch: "<<branch<<endl;
    cout<<"Your CGPA : "<<cgpa<<endl;
    }
};
int main()
{
    Student s1,s2;
    s1.getdata();
    s1.printdata();
    s2.getdata();
    s2.printdata();
```

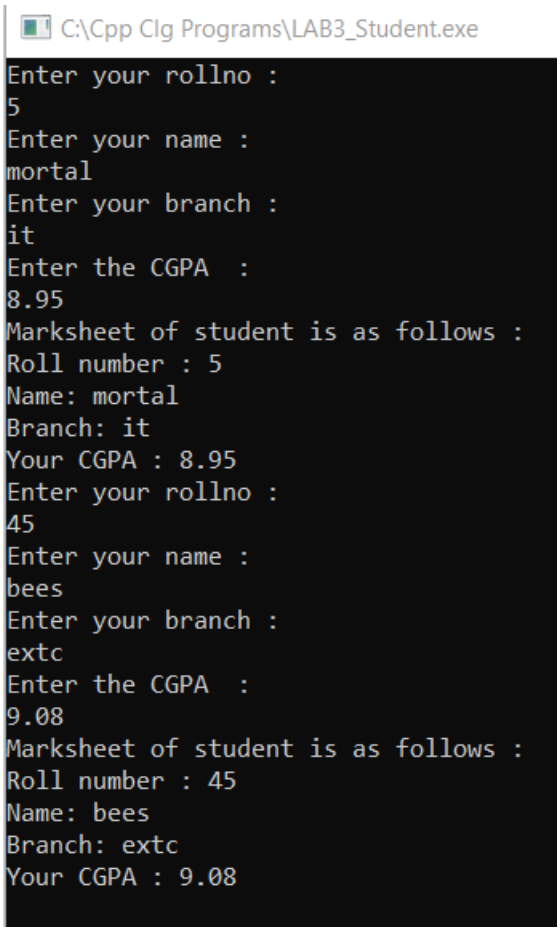
```
    return 0;

}
```

Input given:

For 2 Students information to be output (as declare in code):

1. **Enter roll number**
2. **Enter name**
3. **Enter branch**
4. **Enter your score in (cgpa)**

Output Screenshot:



PROGRAM 2: Write a C++ program for Basic bank Management System.

Algorithm:

Step 1: START

Step 2: Create class bankcode, declare parameters which are needed as name, account number etc.

Step 3: Create Constructor, and functions like deposit(), withdraw(), display().

Step 4: In main function declare objects, and add do-while loop and in it use switch case to add input from user, add input option from given 3.

Step 5: END.

Program code:

```
//To create basic bank management system
#include<iostream>
using namespace std;
class Bankcode
{
public:
string name;
char account_type;
int account_number;
int amount;
float balance;

Bankcode(string n ,int a, char t, float b)
{
        name = n;
        account_number=a;
```

```
        account_type=t;
        balance=b;
}


    int deposit()
        {
            cout<<"Enter the amount to    deposit: "<<endl;
            cin>> amount;
            if(amount<0)
            {
                cout<<"Invalid amount,Enter a valid amount"<<endl;
                return 0;
            }
            balance=balance+amount;
            return 1;
        }
        int withdraw()
        {
            cout<<"Your Balance= "        <<balance;
            cout<<"Enter amount to withdraw: "<<endl;
            cin>> amount;
             if (balance<amount)
            {
                cout<<"Insufficient Balance:  "<<endl;
                return 0;
            }
            if(amount<0)
```

```
                {
                        cout<<"Invalid          amount"<<endl;
                        return 0;
                }
                balance=balance-amount;
                return 1;
        }void display()
        {
                cout<<"Name : "<<name<<endl;
                cout<<"Account Number:" <<account_number<<endl;
                cout<<"Account Type:" <<account_type<<endl;
                cout<<"Balance: " <<balance<<endl;
        }
};
int main()
{
        char ans;
        int account_number;
        int op;
       Bankcode b1("salman",1,'s',2000);
        Bankcode b2("makarand",2,'s',2000);
        Bankcode b3("siddharth",3,'s',2000);
        cout<<"Menu"<<endl;
        cout<<"1.Deposit"<<endl ;
        cout<<"2.Withdraw"<<endl;
        cout<<"3.Display"<<endl;
   cout<<"Enter option"<<endl;
   cin>>op;
```

```
    do{
cout<<"Please enter your account number:"<<endl;
cin>>account_number;
switch(account_number)

{

case 1: if(op==1)
b1.deposit();
if(op==2)
b1.withdraw();
if(op==3)
b1.display();
break;
case 2: if(op==1)
b2.deposit();
if(op==2)
b2.withdraw();
if(op==3)
b2.display();
break;
case 3: if(op==1)
b3.deposit();
if(op==2)
b3.withdraw();
if(op==3)
b3.display();
break;
```

```
default:
cout<<"Enter value between 1 to 3"<<endl;
break;
}
cout<<"Do you want to continue?[Y/N]";
cin>>ans;
if(ans=='Y' || ans == 'y')
{
cout<<"Menu"<<endl;
cout<<"1.Deposit"<<endl;
cout<<"2.Withdraw"<<endl;
cout<<"3.Display"<<endl;
cout<<"Enter option"<<endl;
cin>>op;
}
}
while(ans!='N');
return 0;
}
```

INPUT:

1. **Enter option from 1-3.**
2. **Enter the display amount, withdraw amount.**
3. **Enter y/n.**

Output Screenshot: