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:	**Abhishek Manik Waghmare**
Roll No :	**01**

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:

ECL304.1	Understand object-oriented programming concepts and implement using C++ and Java

Theory:

Q1-Difference between procedural and object oriented language

PROCEDURAL ORIENTED PROGRAMMING	OBJECT ORIENTED PROGRAMMING
In procedural programming, program is divided into small parts called functions.	In object oriented programming, program is divided into small parts called objects.
Procedural programming follows top down approach.	Object oriented programming follows bottom up approach.
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.

Faculty: Ms. Deepali Kayande

Procedural programming does not have any proper way for hiding data so it is *less secure*.	Object oriented programming provides data hiding so it is *more secure*.
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on *unreal world*.	Object oriented programming is based on *real world*.
Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

Q2-Application of object orientation

1. Client-Server Systems

Object-oriented client-server systems provide the IT infrastructure, creating Object-Oriented Client-Server Internet (OCSI) applications. Here, infrastructure refers to operating systems, networks, and hardware. OSCI consist of three major technologies:

- The Client Server
- Object-Oriented Programming
- The Internet

2. Object-Oriented Databases

- These databases try to maintain a direct correspondence between the real-world and database objects in order to let the object retain its identity and integrity. They can then be identified and operated upon.

3. Real-Time System Design

- Real-time systems inherent complexities that make it difficult to build them. Object-oriented techniques make it easier to handle those complexities. These techniques present ways of dealing with these complexities by providing an integrated framework, which includes schedulability analysis and behavioral specifications.

4. Simulation and Modeling System

- It's difficult to model complex systems due to the varying specification of variables. These are prevalent in medicine and in other areas of natural science, such as ecology, zoology, and agronomic systems. Simulating complex systems requires modeling and understanding interactions explicitly. Object-oriented programming provides an alternative approach for simplifying these complex modeling systems.

5. Hypertext and Hypermedia

- OOP also helps in laying out a framework for hypertext. Basically, hypertext is similar to regular text, as it can be stored, searched, and edited easily. The only difference is that hypertext is text with pointers to other text as well.
- Hypermedia, on the other hand, is a superset of hypertext. Documents having hypermedia not only contain links to other pieces of text and information but also to numerous other forms of media, ranging from images to sound.

Q3-Brief introduction to C++ and Java

1) C++

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. C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the ***features & key-points*** to note about the programming language are as follows:

- **Simple**: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
- **Mid-level language**: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
- **Object-Oriented**: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
- **Compiled Language**: C++ is a compiled language, contributing to its speed.

2) JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This tutorial gives a complete understanding of Java. This reference will take you through simple and practical approaches while learning Java Programming language.

Java is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

ALGORITHM:

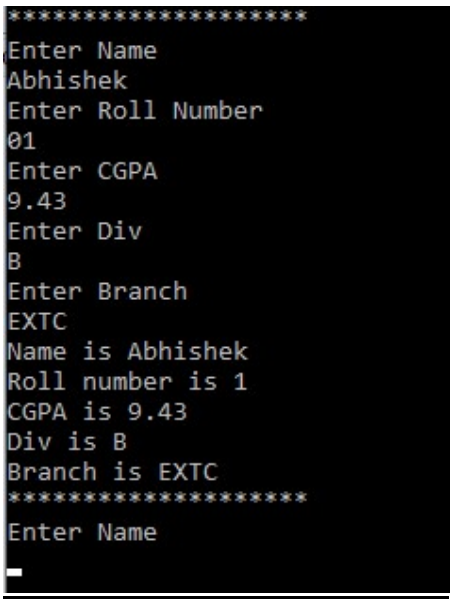
1. Start
2. Define Class Student
3. Define attributes – name, rollnum, cgpa, div, branch
4. Define and declare method – getdata() to get input from user
5. Define and declare method – printdata() to print the values
6. Define Main function()
7. Create object s1, s2 to call the class functionality.
8. End

PROGRAM [3.1]:

```
#include<iostream>

using namespace std;

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

        void getData(){
                cout<<"********************\nEnter Name"<<endl;
                cin>>name;
                cout<<"Enter Roll Number"<<endl;
                cin>>rollnum;
                cout<<"Enter CGPA"<<endl;
                cin>>cgpa;
                cout<<"Enter Div"<<endl;
                cin>>div;
                cout<<"Enter Branch"<<endl;
                cin>>branch;
```

```
        }

        void printData(){
                cout<<"Name is "<<name<<endl;
                cout<<"Roll number is "<<rollnum<<endl;
                cout<<"CGPA is "<<cgpa<<endl;
                cout<<"Div is "<<div<<endl;
                cout<<"Branch is "<<branch<<endl;
        }
};

        int main()
        {
         Student s1,s2;
         s1.getData();
         s1.printData();
         s2.getData();
         s2.printData();
         return 0;
        }
```

OUTPUT [3.1]:



PROGRAM [3.2]:

```
#include<iostream>
using namespace std;
class Bank{
        public:
                string name,address,acc_type;
                long long int mobnum;
                int age;
                float bal = 0,amt,withdrawamt;
```

```
        CreateAccount(){
                cout<<"---------Enter Your Details:---------" <<endl;
                cout<<"Enter your Name: "<<endl;
                cin>>name;
                cout<<"Enter Address: "<<endl;
                cin>>address;
                cout<<"Enter your mobile number: "<<endl;
                cin>>mobnum;
                cout<<"Enter Your age: "<<endl;
                cin>>age;
                cout<<"Which type of account you want to create(savings or current) " <<endl;
                cin>>acc_type;
                cout<<"Your Account is successfully Created!!!!!! "<<endl;
        }

        void deposit(){
                cout<<"Enter the amount to deposit: \nRs";
                cin>>amt;
                bal= bal + amt;
                cout<<"Your amount is successfully deposited!!!!"<<endl;
                cout<<"YOUR BALANCE is Rs"<<bal<<endl;
        }

        void withdraw(){
                cout<<"Enter the amount to withdraw: \nRs"<<endl;
                cin>>withdrawamt;
                if(withdrawamt>bal){
                        cout<<"!!!INSUFFICIENT BALANCE!!!";
                }
                else
                        bal = bal-withdrawamt;
                   cout<<"YOUR BALANCE is Rs"<<bal<<endl;
        }

        void display(){
                cout<<"YOUR BALANCE is Rs" <<bal<<endl;
        }
};
        int main(){
                int choice;
                Bank b;
                while(true){
                        cout<<"ENTER YOUR CHOICE:\n 1.Create Account\n 2.Deposit\n
3.Withdraw\n 4.Display Balance\n 5.Exit"<<endl;
                        cin>>choice;
                        switch (choice) {
         case 1:
```

```
        b.CreateAccount();
        break;
    case 2:
        b.deposit();
        break;
    case 3:
        b.withdraw();
        break;
    case 4:
        b.display();
        break;
    case 5:
            exit(0);
            break;
    default:
            cout<<"Enter valid choice!!!!!";
        break;
                    }
            }
    }
```

OUTPUT [3.2]:

```
ENTER YOUR CHOICE:
 1.Create Account
 2.Deposit
 3.Withdraw
 4.Display Balance
 5.Exit
1
---------Enter Your Details:---------
Enter your Name:
Abhishek
Enter Address:
Bhandup
Enter your mobile number:
1234567890
Enter Your age:
19
Which type of account you want to create(savings or current)
Savings
Your Account is successfully Created!!!!!!
```

```
ENTER YOUR CHOICE:
 1.Create Account
 2.Deposit
 3.Withdraw
 4.Display Balance
 5.Exit
2
Enter the amount to deposit:
Rs5000
Your amount is successfully deposited!!!!
YOUR BALANCE is Rs5000
ENTER YOUR CHOICE:
 1.Create Account
 2.Deposit
 3.Withdraw
 4.Display Balance
 5.Exit
3
Enter the amount to withdraw:
Rs
2599
YOUR BALANCE is Rs2401
ENTER YOUR CHOICE:
 1.Create Account
 2.Deposit
 3.Withdraw
 4.Display Balance
 5.Exit
4
YOUR BALANCE is Rs2401
ENTER YOUR CHOICE:
 1.Create Account
 2.Deposit
 3.Withdraw
 4.Display Balance
 5.Exit
5

--------------------------------
Process exited after 187.8 seconds with return value 0
Press any key to continue . . .
```

GITHUB: https://github.com/Abhishek-0809/Skill-Lab-with-OOPM