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:	**4**
Student Name:	**Sanskar Kumar**
Roll No :	**35**

Title:

4.1 Write a Java 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.

4.2 Write a Java 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:

Explain about Constructor.

In class-based object-oriented programming, a constructor (abbreviation: ctor) is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.

A constructor resembles an instance method, but it differs from a method in that it has no explicit return type, it is not implicitly inherited and it usually has different rules for scope modifiers. Constructors often have the same name as the declaring class. They have the task of initializing the object's data members and of establishing the invariant of the class, failing if the invariant is invalid. A properly written constructor leaves the resulting object in a valid state. Immutable objects must be initialized in a constructor.

Explain about classes and objects in Java

CLASS

Class are a blueprint or a set of instructions to build a specific type of object. It is a basic concept of Object-Oriented Programming which revolve around the real-life entities. Class in Java determines how an object will behave and what the object will contain.

OBJECT

Object is an instance of a class. An object in OOPS is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. For example color name, table, bag, barking. When you send a message to an object, you are asking the object to invoke or execute one of its methods as defined in the class.

From a programming point of view, an object in OOPS can include a data structure, a variable, or a function. It has a memory location allocated. Java Objects are designed as class hierarchies.

How to access class attributes and methods? Explain with example

A class is an element in object oriented programming that aggregates attributes(fields) - which can be public accessible or not - and methods(functions) - which also can be public or private and usually writes/reads those attributes.

Accessing Attributes

You can access attributes by creating an object of the class, and by using the dot syntax (.):

The following example will create an object of the Main class, with the name myObj. We use the x attribute on the object to print its value:

Example

Create an object called "myObj" and print the value of x:

```
 public class Main {
 int x = 5;
 public static void main(String[] args) {
   Main myObj = new Main();
   System.out.println(myObj.x);
 }
}
```

METHODS

A method is a block of code which only runs when it is called.

You can pass data, known as parameters, into a method.

Methods are used to perform certain actions, and they are also known as functions.

To call a method in Java, write the method's name followed by two parentheses () and a semicolon;

In the following example, myMethod() is used to print a text (the action), when it is c

Example

Inside main, call the myMethod() method:

```
 public class Main {
 static void myMethod() {
   System.out.println("I just got executed!");
 }
  public static void main(String[] args) {
   myMethod();
 }
}
```

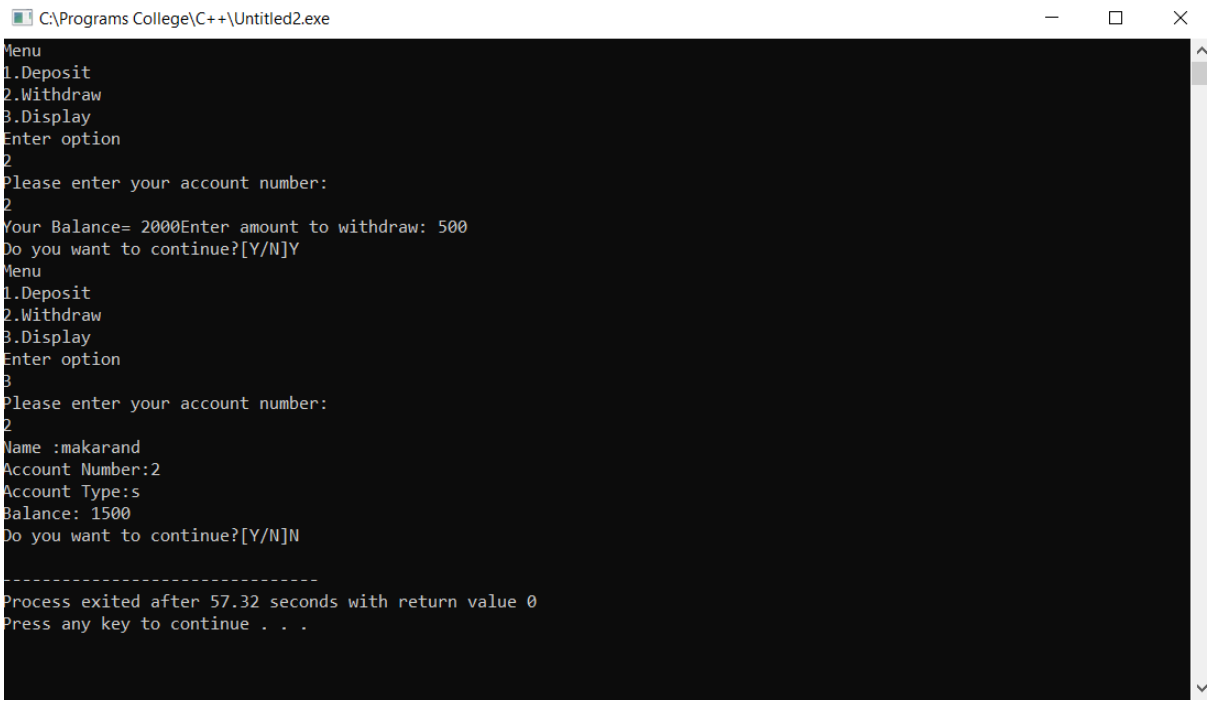
Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Algorithm :	**STEP 1. Start** **STEP 2. Define Class Student** **STEP 3. Define attributes – Name , Roll_no, cgpa, div , branch** **STEP 4. Define and declare method – getdata() to get input from user.** **STEP 5. Define and declare method – printdata() to print the values** **STEP 6. Define Main function()** **STEP 7. Create object s1, s2 to call the class functionality.** **STEP 8. Print result** **STEP 9. End.**
Program:	**https://github.com/SanskarKumar777/Skill-Lab-with-OOPM/commit/402410d8127ca2aea71d2ef379705f91d9ab3c51**
Input given:	**Sanskar** **35** **EXTC** **8.4** **Sam** **23** **EXTC** **9.6**

Faculty: Ms. Deepali Kayande

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

OUTPUT SCREENSHOT:



Algorithm:	**STEP 1. Start** **STEP 2. Define Class BankLab 2** **STEP 3. Define attributes – Name , account_type , account_number,** **amount, balance.**

Faculty: Ms. Deepali Kayande

	STEP 4. Declare attributes by using constructor of class. **STEP 5. Define and declare method – deposit() to deposit the amount** **STEP 6. Define and declare methods – withdraw() to withdraw the amount** **STEP 7. Define and declare methods – display() to display the account details** **STEP 8. Define Main function()** **STEP 9. Create object b1, b2, b3 to call the class functionality.** **STEP 10. Do – while loop to repeat the process.** **STEP 11. Print result** **STEP 12. end**
Program:	**https://github.com/SanskarKumar777/Skill-Lab-with-OOPM/commit/402410d8127ca2aea71d2ef379705f91d9ab3c51**
Input Given:	**Enter option** **1** **Please enter your account number:** **2** **Enter the amount to** **2000** **deposit:** **Do you want to continue? [Y/N]** **Y** **Enter option** **3** **Please enter your account number:** **2** **Do you want to continue? [Y/N]** **N**

OUTPUT SCREENSHOT: