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:	**Prathamesh Ashok Yerekar**
Roll No :	**25**

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:

Q1. Explain about Constructor.

A constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.It is a special type of method which is used to initialize the object.
Every time an object is created using the new() keyword, at least one constructor is called.It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.
There are two types of constructors in Java: no-arg constructor and parameterized constructor.

Q2. Explain about classes and objects in Java

CLASS: It is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these in order:

1. Modifiers: A class can be public or has default access class keyword: class keyword is used to create a class.
2. Class name: The name should begin with an initial letter (capitalized by convention).
3. Superclass (if any): The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
4. Interfaces (if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. Body: The class body surrounded by braces, { }.

OBJECTS: It is a basic unit of Object-Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object basically consists state, behaviour, identity.

Q3.How to access class attributes and methods? Explain with example.

You can access attributes by creating an object of the class, and by using the dot syntax (.)In the example we will create an object of the Main class, with the name myObj. We use the x attribute on the object to print its value.

EG. 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);
 }
}
```

Or override existing values:

EG. Change the value of x=25

```
public class Main {
 int x = 10;
```

```
 public static void main(String[] args) {
   Main myObj = new Main();
   myObj.x = 25; // x is now 25
   System.out.println(myObj.x);
 }
}
```

If you don't want the ability to override existing values, declare the attribute as final:

```
public class Main {
 final int x = 10;

 public static void main(String[] args) {
   Main myObj = new Main();
   myObj.x = 25; // will generate an error: cannot assign a value to a final variable
   System.out.println(myObj.x);
 }
}
```

METHODS: Methods define behaviour of a class. A method contains business logic which is executed when the method is invoked. Methods are the ways to manipulate objects data. Let's take a look at the below example

Syntax of a method :

<access-modifier> <return-type> <name-of-the-method> ({optional}<type-of-parameter> <name-of-the-parameter>){ //method logic //method logic }

example: public void displayPlayerInfo()

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.

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:

```
import java.util.Scanner;
 class Student
{
    Scanner in=new Scanner(System.in);
    String name;
    int rollno;
    String department;
    float cgpa;

//method overloading
 void getData()
{
    Scanner t = new Scanner(System.in);
System.out.println("Student name =");
name= t.next();
System.out.println("Student rollno =");
rollno= t.nextInt();
System.out.println("Student department =");
department= t.next();
System.out.println("Student cgpa=");
cgpa= t.nextFloat();
}

 void getdata( String n, int r, String d, float c)
 {
        name=n;
        rollno=r;
        department=d;
        cgpa=c;
 }

 void printdata()
```

```
 {
        System.out.println("Student name =" +name);
        System.out.println("Student rollno =" +rollno);
        System.out.println("Student department ="+department);
        System.out.println("Student cgpa ="+cgpa);
 }
};
public class StudentTest
{
public static void main(String args[])
{
Student s1=new Student();
Student s2=new Student();

s1.getData(); //non parameter
s1.printdata();

s2.getdata("ram", 25, "EXTC", (float)9.8 );
s2.printdata();

}
}
```

Input given:

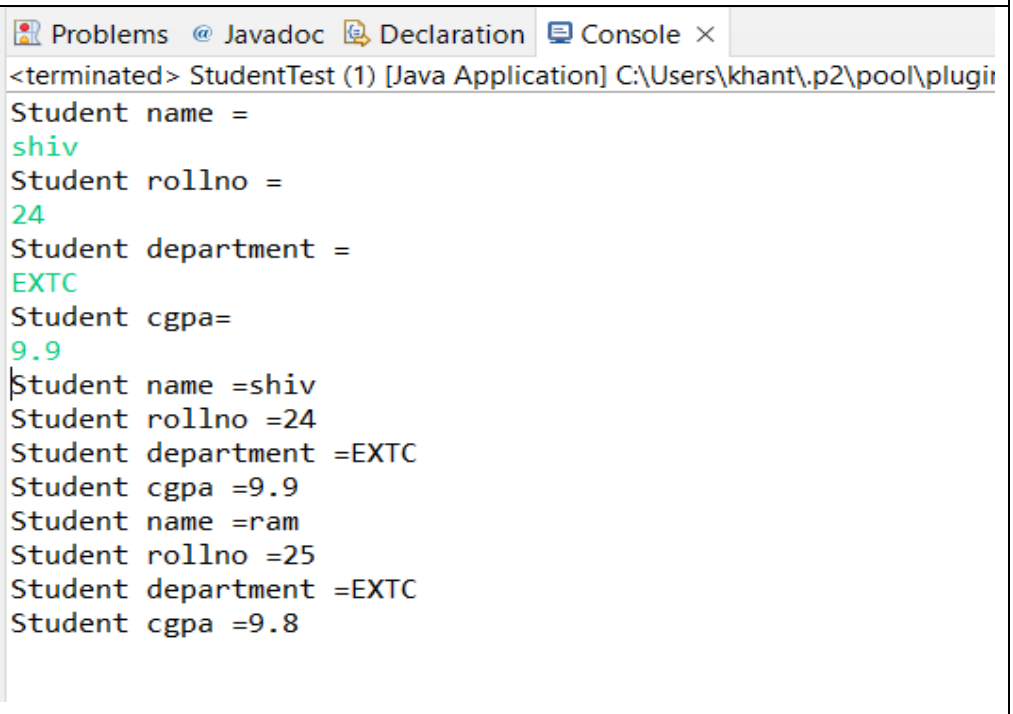
Student name= shiv

Student rollno=24

Student department= EXTC

Student cgpa=9.9

Output Screenshot:



2. Write a Java program for Basic bank Management System

Algorithm :	STEP 1. Start STEP 2. Define Class BankLab 2 STEP 3. Define attributes – Name , account_type , account_number, amount, balance \ 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 results STEP 12. end
Program:	`import java.util.Scanner;` `public class BankLab2 {` `Scanner in=new Scanner(System.in);` `String name;` `char account_type;` `int account_number,amount;` `float balance;` `public BankLab2(String n,int a, char t, float b) {` `// TODO Auto-generated constructor stub` `name = n;`

```
		account_number=a;
		account_type=t;
		balance=b;


	}


	int deposit()
	{
		System.out.println("Enter the amount to deposit:
");
		int amount=in.nextInt();
		if(amount<0)
		{
			System.out.println("Invalid amount,Enter a
valid amount");
			return 0;
		}
		balance=balance+amount;
		return 1;
	}


	int withdraw()
	{
		System.out.println("Your Balance= "	+balance
);
		System.out.println("Enter amount to withdraw:
");
		int amount=in.nextInt();
		if (balance<amount)
```

```
		{
			System.out.println("Insufficient Balance: ");
			return 0;
		}
		if(amount<0)
		{
			System.out.println("Invalid amount" );
			return 0;
		}
		balance=balance-amount;
		return 1;
	}

	void display()
	{
		System.out.println("Name :"+name);
		System.out.println("Account Number:" +account_number);
		System.out.println("Account Type:" +account_type);
		System.out.println("Balance: " +balance);
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner in=new Scanner(System.in);
		BankLab2 b1=new BankLab2("salman",1,'s',2000);
		BankLab2 b2=new BankLab2("makarand",2,'s',2000);
```

```
                BankLab2 b3=new
BankLab2("siddharth",3,'s',2000);



                System.out.println("Menu");

                System.out.println("1.Deposit");

                System.out.println("2.Withdraw");

                System.out.println("3.Display");

                System.out.println("Enter option");

                int op=in.nextInt();

                char ans;

                do

                {

                        System.out.println("Please enter your
account number:");

                        int account_number=in.nextInt();

                                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:
System.out.println("Enter value between 1 to 3");
                                                break;
                                }
                                System.out.println("Do you want to
continue?[Y/N]");
                                ans=in.next().charAt(0);  //char
input in variable ans
                                if(ans=='Y' || ans == 'y')
                                {
                                        System.out.println("Menu");

        System.out.println("1.Deposit");
```

	`` System.out.println("2.Withdraw"); System.out.println("3.Display"); System.out.println("Enter option"); op=in.nextInt(); } } while(ans!='N'); } } ```
Input given:	Entered option=2 Entered account number=2 Amount to withdraw=100 Continue Entered option=2 Entered account number=2

Output Screenshot: