Experiment No.06

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim: **To Study Constructors, static and Overloading in java.**

Q1: Given a diagram representing computational problem, initialize the data members using parameterized constructor.

In the retail application as customer is registered to retail shop, so the details of the customer must also be initialized Print the details of the customer.

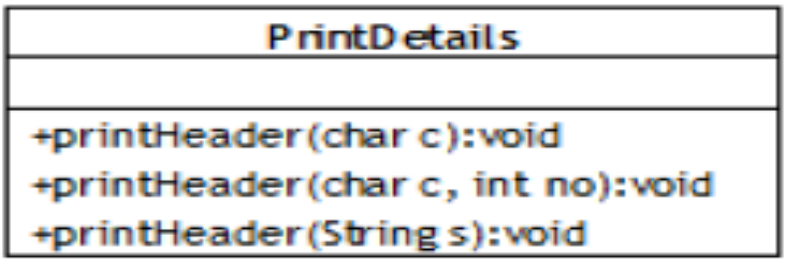
Customer
-customerId : int -telephoneNo : long[] -customerName : String
+ Customer(int, long[], String) + setCustomerId(int) : void +getCustomerId() : int +setTelephoneNo(long[]) : void +getTelephoneNo() : long[] +setCustomerName(String) : void +getCustomerName() : String +validateCustomerName() : boolean

Q 2 In the retail application, each time a customer is registered, customer id must be automatically generated starting from 1001. Also, retail shop management wants to know how many customers have registered at a point of time.

Customer
-customerId : int
+Customer() +setCustomerId(int) : void +getCustomerId() : int +totalNoOfCustomers() : int

Q 3 Retail store has a requirement for printing many reports including the bills. All reports contain header. The header may be

1) A line containing a character printed 70 times or
2) A title of a report or
3) A line containing a character specified number of times



A.2 Prerequisite:

C++ programing with constructors and static

A.3 Outcome:

After successful completion of this experiment students will be able to understand the constructors its different types and Static variables, methods and Blocks

A.4 Theory:

Constructors

Constructor in java is a *special type of method* that is used to initialize the object. Java constructor is *invoked at the time of object creation*. It constructs the values i.e. provides data for the object that is why it is known as constructor.

Rules for creating java constructor

There are basically two rules defined for the constructor.

1. Constructor name must be same as its class name
2. Constructor must have no explicit return type

There are two types of constructors:

1. Default constructor (no-arg constructor)
2. Parameterized constructor

Default Constructor

```
1. class Bike1{
2. Bike1(){System.out.println("Bike is created");}
3. public static void main(String args[]){
4. Bike1 b=new Bike1();
```

```
5.  }
6.  }
```

Parameterized Constructor

```
1.  class Student4{
2.      int id;
3.      String name;
4.
5.      Student4(int i,String n){
6.      id = i;
7.      name = n;
8.      }
9.      void display(){System.out.println(id+" "+name);}
10.
11.     public static void main(String args[]){
12.     Student4 s1 = new Student4(111,"Karan");
13.     Student4 s2 = new Student4(222,"Aryan");
14.     s1.display();
15.     s2.display();
16.    }
17. }
```

STATIC

The **static keyword** in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class.

The static can be:

1. variable (also known as class variable)
2. method (also known as class method)
3. block

Static Variables

If you declare any variable as static, it is known static variable.

The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees,college name of students etc.

The static variable gets memory only once in class area at the time of class loading.

class Counter2{

static int count=0;//will get memory only once and retain its value

```
Counter2(){

count++;

System.out.println(count);

}

public static void main(String args[]){

Counter2 c1=new Counter2();

Counter2 c2=new Counter2();

Counter2 c3=new Counter2();

 }

}
```

Static Method

If you apply static keyword with any method, it is known as static method.

- A static method belongs to the class rather than object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- static method can access static data member and can change the value of it.

```
class Calculate{
static int cube(int x){
return x*x*x;
}
public static void main(String args[]){
int result=Calculate.cube(5);
System.out.println(result);
}  }
```

Static Block

- Is used to initialize the static data member.
- It is executed before main method at the time of classloading.

```
class A2{

  static{System.out.println("static block is invoked");}

  public static void main(String args[]){

    System.out.println("Hello main");  }}
```

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No.	Name:
Program:	Division:
Semester:	Batch :
Date of Experiment:	Date of Submission:
Grade :	

B.1 Program Code

Q1)

```
package exp_6;


import java.util.*;

class Customer {
    private int customerId;
    private long telephoneNum;
    private String customerName;

    public Customer(int id, long num, String name) {
        this.customerId = id;
        this.telephoneNum = num;
        this.customerName = name;

    }

    public void setCustomerId(int id) {
        this.customerId = id;
    }

    public int getCustomerId() {
        return this.customerId;
    }

    public void setCustomerName(String name) {
```

```
        this.customerName = name;
    }

    public String getCustomerName() {
        return this.customerName;
    }

    public void setTelephoneNum(long num) {
        this.telephoneNum = num;
    }

    public long getTelephoneNum() {
        return this.telephoneNum;
    }

    public boolean validateCustomerName() {
        return this.customerName.matches("[A-Z][a-z]*");
    }
}
```

Q2)

```
package exp_6;


class CustomerQ2 {
    private int customerId, total = 0;

    public void setCustomerId(int id) {
        this.customerId = 1000 + (++total);
    }

    public int getCustomerId() {
        return this.customerId;
    }
}


package exp_6;


class PrintDetails {
    public void printHeader(char c){
        for(int i=0 ; i<70 ; i++)
            System.out.println(c);
```

```
    }

    public void printHeader(char c, int num){
        for(int i=0 ; i<num ; i++)
            System.out.println(c);
    }

    public void printHeader(String str){
            System.out.println(str);
    }
}
```

B.2 Output

B.3 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

B.4Question of Curiosity:

Q1) Can we overload main method?

Q2) Can we Copy the value of one object to another in java? Explain with example