PRACTICAL – 1.1

AIM:-Write a program to count the number of words that start with a capital letter.

INPUT:-

```
import java.util.*;
public class
CountCapitalLetter{ public
static void main(String args[]){
String s1;
Scanner sc=new
Scanner(System.in);
System.out.println("Enter the
String :"); s1=sc.nextLine(); int
count=0,i=0,n; n=s1.length();
System.out.println("Size of the String is :"+n);
if(null==s1 || s1.isEmpty()){
System.out.println("Text Empty");
}
else{
if(Character.isUpperCase(s1.charAt(
0)))
{
count++;
}
for(i=1;i<n;i++)
```

```
{
if(Character.isWhitespace(s1.charAt(i-1)) && Character.isUpperCase(s1.charAt(i)))
 {
count++;
}
}
}
System.out.println("Number of Word which Start With Capital Letter :"+count);
}
}
```

OUTPUT

```
Enter the String :
Govind Jha
Size of the String is :10
Number of Word which Start With Capital Letter :2

Process finished with exit code 0
```

PRACTICAL – 1.2

AIM:- Write a java program to take an array of strings as an input, and arrange strings in ascending order.

INPUT:-

```
package  com.company;
import java.util.Scanner;
public class Main {

    public static void main(String[] args) {
        int n;
        String temp;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Number of String you would like to Enter :"); n=sc.nextInt();

        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(str[i].compareTo(str[j])>0)
                {
                    temp=str[i]; str[i]=str[j];

                    str[j]=temp;
                }
            }
        }
        System.out.println("Strings in Sorted Order :"); for(int i=0;i<n;i++)
        {
            System.out.print(str[i]+" ");
        }
    }
}
```

OUTPUT

```
Enter Number of String you would like to Enter :1 2 3 4 5 6
Enter the String one by one :
1 3 2 4 5 6
Strings in Sorted Order :
1 3 2 4 5 6
Process finished with exit code 0
```

PRACTICAL – 2.1

AIM:- Write a program to find the largest number in an array of numbers using command line arguments.

INPUT:-

```
package  com.company;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int n;
        int temp;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements you want to store: ");
        n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        System.out.println("Array elements are: ");
        for (int i = 0; i < n; i++) {
            System.out.println(array[i]);
        }
        for (int i = 0; i < array.length; i++) {
            for (int j = i + 1; j < array.length; j++) {
                if (array[i] < array[j]) {

                    temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
            }
        }

        System.out.println();

        System.out.println("Largest value is : "+array[0]);

    }
}
```

OUTPUT

```
Enter the number of elements you want to store: 5
Enter the elements of the array:
1
5
6
9
4
Array elements are:
1
5
6
9
4

Largest value is : 9

Process finished with exit code 0
```

PRACTICAL – 2.2

AIM:- Write a program to find factorial of number. Here, take number as command line argument.

INPUT:-

```
package com.company;

public class Main {

        public static void main(String args[]){
            int i,fact=1;
            int number=3;
            for(i=1;i<=number;i++){

                fact=fact*i;
            }
            System.out.println("Factorial of "+number+" is: "+fact);
        }

    }
```

OUTPUT

```
Factorial of 3 is: 6

Process finished with exit code 0
```

PRACTICAL – 3.1

AIM:- Write a program to demonstrate class and objects using the concept of an array object.

INPUT:-

```
package  com.company;

import java.util.Scanner;

class Student{
    String Name;
    int Enroll;
    int[] array = new int[4];
    void getData()
    {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter students name : ");
        Name = ob.nextLine();
        System.out.println("Enter Enrollment no. :");
        Enroll = ob.nextInt();
        System.out.println("Enter Fees paid per year :");

        for (int i = 0;i<4;i++)
        {
            array[i]=ob.nextInt();
        }
    }
    void display()
    {
        int total = 0;
        System.out.println("fees paid are : ");
        for (int i = 0;i<4;i++) {
            System.out.println(+array[i]);

        }
        for (int i=0;i<4;i++) {
            total = total + array[i];

        }
        System.out.println("Total Fees paid are : "+total);

    }
}
public class Main {
```

```
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("How Many Students :");
        int n=sc.nextInt();
        Student[] abc=new Student[n];
        for(int i=0;i<n;i++)
        {
            abc[i]=new Student();
            abc[i].getData();
            abc[i].display();
        }

    }
}
```

OUTPUT

```
How Many Students :
1
Enter students name :
Govind Jha
Enter Enrollment no. :
39
Enter Fees paid per year :
1
2
3
4
fees paid are :
1
2
3
4
Total Fees paid are : 10

Process finished with exit code 0
```

PRACTICAL – 3.2

AIM:- . Declare a class Box. Overload Box constructors with zero argument, one argument and three Argument to initialize the members of the class. Declare a method to find volume of the box.

INPUT:-

```
package com.company;
import java.util.Scanner;

class Box
{
   double length,height,radious;
   void getArea()
   {
      double Area,pi=3.14;
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Radious of Circle : ");
      radious = s.nextDouble();
      Area = 2*pi*radious;
      System.out.println("Area Of Circle = "+Area);

   }
   void getVolume()
   {
      double Volume;
      System.out.println("Enter Length & Height of box : ");
      Scanner s1 = new Scanner(System.in);
      length=s1.nextDouble();
      height= s1.nextDouble();
      Volume = length*height;
      System.out.println("Volume of box = "+Volume);
   }
}
public class Main {

   public static void main(String[] args) {
      Box abc=new Box();
      Box abc1=new Box();
         abc.getArea();
         abc1.getVolume();


   }
}
```

OUTPUT

```
Enter Radious of Circle :
2
Area Of Circle = 12.56
Enter Length & Height of box :
2
2
Volume of box = 4.0

Process finished with exit code 0
```

PRACTICAL – 4.1

AIM:- Write a program to demonstrate garbage collection using System.gc() or Runtime.gc().

INPUT:-

```
package com.company;
class Garbage{
   public void finalize(){System.out.println("object is garbage collected");}
   public static void main(String args[]){
      Garbage s1=new Garbage();
      Garbage s2=new Garbage();
      s1=null;
      s2=null;
      System.gc();
   }
}
```

OUTPUT

```
object is garbage collected
object is garbage collected

Process finished with exit code 0
```

PRACTICAL – 4.2

AIM:- Write a program to show the use of finalize method for garbage collection.

INPUT:-

```
package com.company;
class Garbage {

    public static void main(String[] args)
    {
        Garbage obj = new Garbage();
        System.out.println(obj.hashCode());
        obj = null;
        System.gc();
        System.out.println("end of garbage collection");

    }
    @Override
    protected void finalize()
    {
        System.out.println("finalize method called");
    }
}
```

OUTPUT

```
245257410
end of garbage collection
finalize method called

Process finished with exit code 0
```

PRACTICAL – 5

AIM:- Write a program to demonstrate static constants and final constants.

INPUT:-

```
package com.company;

class sconst
    {
       final float pi=3.14f; static int r=3; static int b=4; static int c; static void display()
    {
       System.out.println("Value Of r is:" );
    }
       static void add()
       {
          c=r+b;
          System.out.println(c);
       }
       public static void main(String args[])
       {
          sconst obj=new sconst(); obj.display(); add();
          System.out.println(sconst.r);
       } }
```

OUTPUT

```
Value Of r is:
7
3

Process finished with exit code 0
```

PRACTICAL – 6

AIM:- Write a program to explain static polymorphism in java.

INPUT:-

```
package com.company;

class Bike{

    void run()

    {

        System.out.println("running");

    }

}

class Splender extends Bike {

    void run() {

        System.out.println("running safely with 60km");

    }

}

class Activa extends Bike {

    void run() {

        System.out.println("running safely with 70km");

    }

}


public class Main {

    public static void main(String args[]) {

        Bike b = new Splender();

        Bike c = new Activa();

        b.run();
```

```
        c.run();
    }
}
```

OUTPUT

```
running safely with 60km
running safely with 70km

Process finished with exit code 0
```

PRACTICAL – 7.1

AIM:- Write a program to find the factorial of a number using interface.

INPUT:-

```
package com.company;


import java.util.Scanner;
        interface fact
       {
           void show_fact(int n);
       }
      class Factorial implements fact
       {
           public void show_fact(int n)
           {
               int fact=1;
               for(int i=1;i<=n;i++)
               {
                   fact=fact*i;
               }
               System.out.println("Factorial of number is :"+fact);
           }
           public static void main(String args[])
           {
               System.out.println("Enter Number for Factorial : ");
               Scanner s = new Scanner(System.in);
```

```
        int n = s.nextInt();
        Factorial ob=new Factorial();
        ob.show_fact(n);
    }
}
```

OUTPUT

```
Enter Number for Factorial :
5
Factorial of number is :120

Process finished with exit code 0
```

PRACTICAL – 7.2

AIM:- Write a program to implement multiple inheritance in java using interface.

INPUT:-

```
package com.company;

    interface AnimalEat {
        void eat();
    }
    interface AnimalTravel {
        void travel();
    }
    class Animal implements AnimalEat, AnimalTravel {
        public void eat() {
            System.out.println("Animal is eating");
        }
        public void travel() {
            System.out.println("Animal is travelling");
        }
    }
    public class Main {
        public static void main(String args[]) {
            Animal a = new Animal();
            a.eat();
            a.travel();
        }
    }
```

OUTPUT

```
Animal is eating
Animal is travelling


Process finished with exit code 0
```