Ambekeshwar Group of Institutions

Technology & Management, Lucknow

Practical File

Branch :- CSE 2nd Year

Subject :- Object oriented programming using JAVA

Name:- Suraj Arya

Date:-

.....

Teacher In-charge

.....

Examiner's Signature

.....

Principal

Institution Stamp

PROGRAM = "1"

Ques:- WAP to show matrix.

```
class A{
        Public void point_A() {System.out.println
                ("class A"); }
}
Class B extends A {
        Public void print_B() {System.out.println
                ("class B"); }
}
Class C extends A {
        Public void print_C() {System.out.println
                ("class C"); }
}
Class D extends A {
        Public void print_D() {System.out.println
                ("class D"); }
}
// Driver class
public class test {
        public static void main (String [] args)
{
```

```
        B obj_B = new B ();
        obj_B. print_A ();
        obj_B. print_B ();

        C obj_C = new C ();
        obj_C. print_A ();
        obj_C. print_C ();

        D obj_D = new D ();
        obj_D. print_A ();
        obj_D. print_D ();
    }
}
```

* Output :-

```
Class A
Class B
Class A
Class C
Class A
Class D
```

PROGRAM="2"

Ques :- WAP For factorial of a number.

```
Class Factorial {
int Factorial (int n)
{
        // Single line to find factorial
        return (n==1 || n==0) ? 1 : n * Factorial (n-1);
}
// Driver Code
Public Static Code main (String args[])
{
   Factorial obj = new Factorial ();
     int num = 6;
   System.out.println ("Factorial of" + num +
                        "is" + obj.Factorial (num));
}
}
```

* Output :-

Factorial of 6 is 720

PROGRAM = "3"

Ques :- WAP to handle the exception using try and multiple Catch block.

```
import java.util.Scanner;

Public class exce
{
    public Static void main (Strings args[])
    {
        Scanner Scn = new Scanner (System.in);
        try
        {
            int n = Integer.parseInt (scn.next line());
            if (98% n==0)
                System.out.println (n+ "is a factor of
                                    98");
        }
        Catch (Number Format Exception | Arithmetic Exception
                                    ex)
        {
            System.out.println ("Exception encountered"
                                    + ex );
        }
```

}

}

* Output :-

Exception encountered java.lang.ArithmeticException:
/ by Zero

PROGRAM = "4"

Ques :- WAP to show String reverse.

```
import java.lang.*;
import java.io.*;
import java.util.*;
// Class of Reverse String
Class Reverse String {
        Public Static void main (String [] args)
        {
            String input = "Puneet";

            // get Bytes () method to convert string
            // into bytes [].
            byte [] Str As Byte Array = input.get Bytes ();

            byte [] result = new byte [Str As Byte Array.length];

            // Store result in reverse order into the
            // result byte []
            For (int i=0; i < Str As Byte Array.length; i++)
                result [i]= Str As Byte Array
                        [Str As Byte Array.length-i-1];

            System.out.println (new String (result));
        }
}
```

* Output :-

teezu P

PROGRAM = "5"

Ques :- WAP to show matrix.

```
import java.io.*;
import java.util.*;
 Class matrix {
        public static void main (int mat[][])
        {
              // Loop through all rows
              for (int[] row : mat)
                    // Converting each row as String
                    // and then printing in a Separate
                       line
                    System.out.println (Array.toString (row));
        }
       Public static void main (String args[])
                    throws IO Exception
       {
              int mat [][] = {{ 6,7,9,4 },
                              {5,6,5,8},
                               {9,15,11,19}};
       }
}
```

* Output :-

[6, 7, 9, 4]

[5, 6, 5, 8]

[9, 15, 11, 19]

PROGRAM = "6"

Ques :- WAP to Show Even and Odd.

```
import java.util.*;
// Main class
// Test Even Odd by checking LSB
Public class GFG {

    //Method 1
    // To test number is even or odd
    Public Static String testOddEvenByCheckingLSB
                                        (int a)
    {
        if (a!=0){
        if (integer.toBinaryString(a).endsWith("0"))
            return "Even";
    }           {
        else {
            return "odd";
        } }
    }
    // Here we will land if
    // it does not ends with 0
    else {
            return "Zero";
```

```
    }
}
// Method 2
// Main driver method
Public Static void main (String [] args)
{
        // Iterating over using for loop
        for (int i=0; i<=10; i++){

            // Calling the function and printing
            // Corresponding number is even or odd
            System. out. println (
                i+ ":" + test Odd Even by Checking LSB(i));
        }
    }
}
```

* Output :-

```
0 : zero
1 : odd
2 : even
3 : odd
4 : even
5 : odd
6 : even
7 : odd
8 : even
9 : odd
10 : even
```

PROGRAM = "7"

Ques :- WAP to design a class using abstract methods and Classes.

```
abstract Class MotorBike {
    abstract void brake ();
}
Class SportsBike extends MotorBike {
    // implementation of abstract method
    public void brake () {
        System. out. println ("SportsBike Brake");
    }
}
Class MountainBike extends MotorBike {
    // implementation of abstract Method
    public void brake () {
        System . out. println ("Mountain Bike Brake");
    }
}
Class Main {
    Public Static void main (String [] args) {
        Mountain Bike m1 = new Mountain Bike ();
        m1 . brake ();
```

```
        SportsBike S1 = new SportsBike();
        s1.brake();
    }
}
```

* Output :-

```
Mountain Bike Brake
Sports Bike Brake.
```