

NAME:ROHAN NYATI

SAP ID:500075940

ROLL NO. : R177219148

BATCH-5(AI&ML)

## EXPERIMENT NO – 7

TITLE: Exceptions

**1. Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsException, so that any such problem doesn't cause illegal termination of program.**

### CODE

```
package rohan;
import java.io.*;
class student
{
    String name,grade;
    int reg,m1,m2,m3;
    void read()throws Exception
    {
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter the register no:");
        reg=Integer.parseInt(in.readLine());
        System.out.println("Enter name :");
        name=in.readLine();
        System.out.println("Enter marks 1:");
        m1=Integer.parseInt(in.readLine());
        System.out.println("Enter marks 2:");
        m2=Integer.parseInt(in.readLine());
        System.out.println("Enter marks 3:");
        m3=Integer.parseInt(in.readLine());
    }
    void disp_grade()
    {
        int total=m1+m2+m3;
        if(total>=250) grade="A";
        else if(total>=200) grade="B";
        else if(total>=150) grade="C";
        else if(total>=100) grade="D";
        else grade="E";
        System.out.println("Grade : "+grade);
    }
}
```

```

    }
    void disp()
    {
        System.out.println("Marks of students");
        System.out.println("Register No:"+reg);
        System.out.println("Name :"+name);
        System.out.println("Marks 1:"+m1);
        System.out.println("Marks 2:"+m2);
        System.out.println("Marks 3:"+m3);
        disp_grade();
    }
}

public class Inheritance1
{
    public static void main(String args[])
    {

        int n=0;
        student s=new student();
        try
        {
            DataInputStream in=new DataInputStream(System.in);
            System.out.println("Enter number of students:");
            n=Integer.parseInt(in.readLine());
            for(int x=0;x<n;x++);
                s.read();

        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Maximum no of students is 10\n");
            n=10;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        for(int x=0;x<n;x++);
        s.disp();
    }
}

```

## OUTPUT

---

```
Enter number of students:
1
Enter the register no:
100
Enter name :
Aviral Mehra
Enter marks 1:
90
Enter marks 2:
99
Enter marks 3:
100
Marks of students
Register No:100
Name :Aviral Mehra
Marks 1:90
Marks 2:99
Marks 3:100
Grade : A
```

**2. Write a Java program to enable the user to handle any chance of divide by zero exception.**

## CODE

```
package rohan;
```

```
public class Exception2
{
    public static void main(String ar[])
    {
        int no=0,m=10,result=0;
        try
        {
            result=m/no;
        }
        catch(ArithmeticException e)
        {
            System.out.println(" division by zero ");
            System.out.println(" value of result has been set as one");
            result=1;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println(" Result :"+result);
    }
}
```

```
}
```

## OUTPUT

```
division by zero  
value of result has been set as one  
Result :1
```

**3. Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).**

## CODE

```
package rohan;  
  
import java.io.*;  
  
import java.util.*;  
  
class Numb  
{  
    public static void main(String args[])  
    {  
        int i,j;  
  
        float add,sub,mul,div;  
  
        System.out.println("CALCULATOR:");  
  
        System.out.println("Enter two Operands:");  
  
        Scanner in=new Scanner(System.in);  
  
        try  
        {  
            i=in.nextInt();  
  
            j=in.nextInt();
```

```
add=i+j;

sub=i-j;

mul=i*j;

div=i/j;

System.out.println("Addition =" +add);

System.out.println("Subtraction =" +sub);

System.out.println("Multiplication =" +mul);

System.out.println("Division =" +div);

}

catch(InputMismatchException e)

{

    System.out.println("Program Is Terminated Exception Caught");

}

}

}
```

## OUTPUT

```
CALCULATOR:
Enter two Operands:
50
40
Addition =90.0
Subtraction =10.0
Multiplication =2000.0
Division =1.0
```

---

```
CALCULATOR:
Enter two Operands:
+
Program Is Terminated Exception Caught
```

4. On a single track two vehicles are running. As vehicles are going in same direction there is no problem. If the vehicles are running in different direction there is a chance of collision. To avoid collisions write a Java program using exception handling. You are free to make necessary assumptions.

## CODE

```
package rohan;
import java.io.*;
class collision extends Exception
{
    collision(String s)
    {
        super(s);
    }
}
class coll
{
    public static void main(String args[])
    {
        String t1=null,t2=null;
        try
        {
            DataInputStream in= new DataInputStream(System.in);
            System.out.println("enter the direction of vehicle1:(left/right:");
            t1=in.readLine();
            System.out.println("enter the direction of vehicle2:(left/right:");
            t2=in.readLine();
            if(!t1.equals(t2))
                throw new collision("truck2 has to go on "+ t1 + " direction");
        }
        catch(collision e)
        {
            System.out.println(e);
            t2=t1;
            System.out.println("the collision has been avoided by redirection truck2");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println("direction of truck1 :"+t1);
    }
}
```

```

System.out.println("direction of truck2 :"+t2);
}
}

```

## OUTPUT

```

enter the direction of vehicle1:(left/right):
left
enter the direction of vehicle2:(left/right):
right
aviral.collision: truck2 has to go on left direction
the collision has been avoided by redirection truck2
direction of truck1 :left
direction of truck2 :left

```

5. Write a java program to throw an exception for an employee details. • If an employee name is a number, a name exception must be thrown. • If an employee age is greater than 50, an age exception must be thrown. • Or else an object must be created for the entered employee details

## CODE

```

package rohan;
import java.io.*;
import java.util.*;
class Numx
{
public static void main(String args[])
{
String name;
int age;
System.out.println("-----ENTER EMPLOYEE DETAILS-----");
System.out.println("Enter Name and Age:");
Scanner sc=new Scanner(System.in);
try
{
if(!(sc.nextLine().matches("[a-zA-Z]+")))
{
throw new IOException();
}
age=sc.nextInt();
if(age>50)
{
System.out.println("Age greater than 50 Exception");
throw new Exception();
}
}
}

```

```
Numx x=new Numx();  
    System.out.println("-----Object Created-----");  
}  
catch(Exception e)  
{  
    System.out.println("Exception");  
}  
}  
}
```

## OUTPUT

```
-----ENTER EMPLOYEE DETAILS-----  
Enter Name and Age:  
90  
Exception
```