

R.Anusha

Q1. Write a java program to handle Exception using try, catch, finally block while reading input from commandline and store to integer array.

```
import java.util.*;
class Program1
{
    public static void main(String[] args)
    {
        int[] a = new int[10];

        int i;

        try{
            for(i=0;i<args.length;i++)
            {
                a[i]=Integer.parseInt(args[i]);
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        finally{
            System.out.println("End of program....");
        }
    }
}
```

Output:

```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\iball>F:

F:\>cd "Anudip"

F:\Anudip>javac Exception_Handling.java

F:\Anudip>java Exception_Handling 10 20 30 40
End of program....

F:\Anudip>java Exception_Handling 10 20 DEEPTHI
java.lang.NumberFormatException: For input string: "DEEPTHI"
End of program....

F:\Anudip>java Exception_Handling 10 20 30 40 45 50 550 60 65 70 80 90 100
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 10
End of program....

F:\Anudip>java Exception_Handling 10 20.5826955 30 40 50
java.lang.NumberFormatException: For input string: "20.5826955"
End of program....

F:\Anudip>_
```

Q2. Write a java program for Method level exception handling, for writing data to file using objects.

```
package Lab5;
import java.io.Serializable;
public class Student implements Serializable {

    int idno;
    String Name;

    public Student(int id, String na)
    {
        idno=id;
        Name=na;
    }
}

package Lab5;
import java.io.*;
public class FileWriting {

    public void Writedata() throws Exception
    {
        FileOutputStream fout = new
FileOutputStream("d:\\Test_rec.txt");
        ObjectOutputStream out = new
ObjectOutputStream(fout);
        Student s = new Student(100,"Sam");
        // s.Show();
        out.writeObject(s);

        System.out.println("data written to
file...");
    }

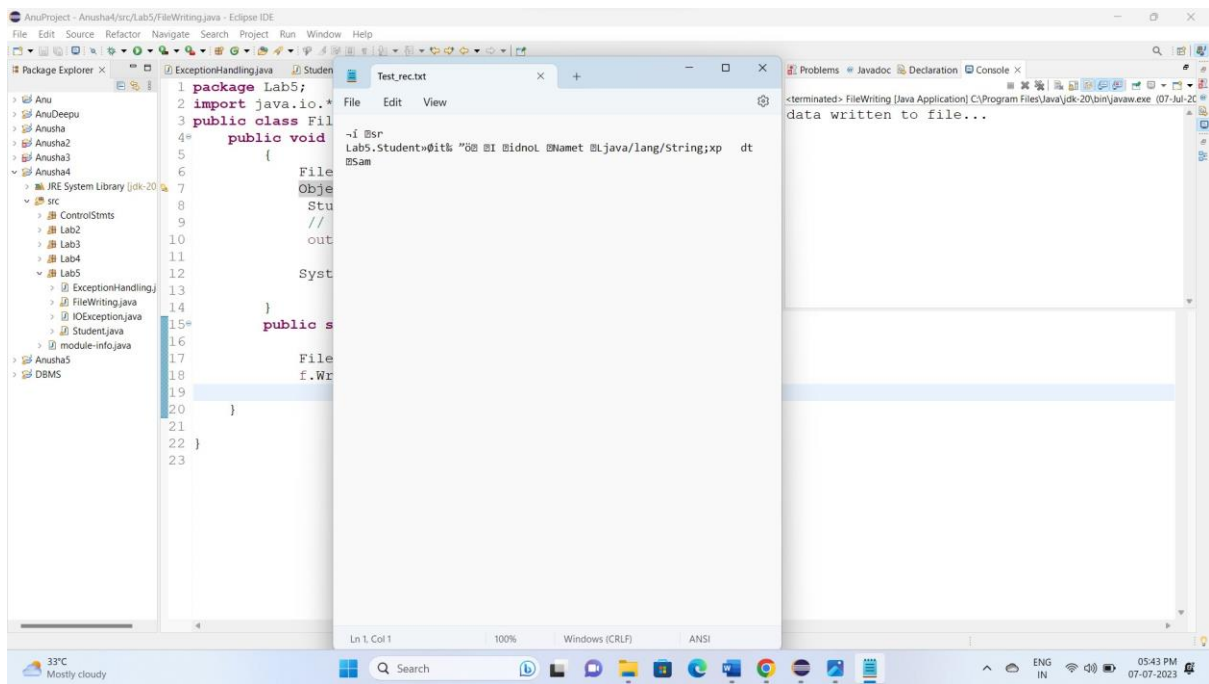
    public static void main(String[] args) throws
Exception {

        FileWriting f = new FileWriting();
        f.Writedata();
    }
}
```

```
}  
}
```

Output:

data written to file...



Q3. Write a java program to illustrate, user can check error condition and call the catch block.

```
package Lab5;
import java.util.*;
public class throwExample {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int a,b,c;
        try {
            System.out.println("Enter 2 integer
values ");
            a=sc.nextInt();
            b=sc.nextInt();

            if(b==0)
            {
                Exception eobj = new
Exception("divisor must be non zero value ");
                throw(eobj); //call the catch block
manually

            }
            else
            {
                c=a/b;
                System.out.println("dvivison "+ c);
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }

    }
}
```

Output:

Enter 2 integer values

20

10

dvivison 2

Q4. Write a java program to illustrate IO exception

```
package Lab5;
import java.util.*;
public class IOException {

    public static void main(String[] args) {

        //Create a new scanner with the specified String
        Object
        Scanner scan = new Scanner("Hello World!
Hello JavaTpoint.");
        //Print the line
        System.out.println("" + scan.nextLine());
        //Check if there is an IO exception
        System.out.println("Exception Output: " +
scan.ioException());
        scan.close();
    }

}
```

Output:

```
Hello World! Hello JavaTpoint.
Exception Output: null
```