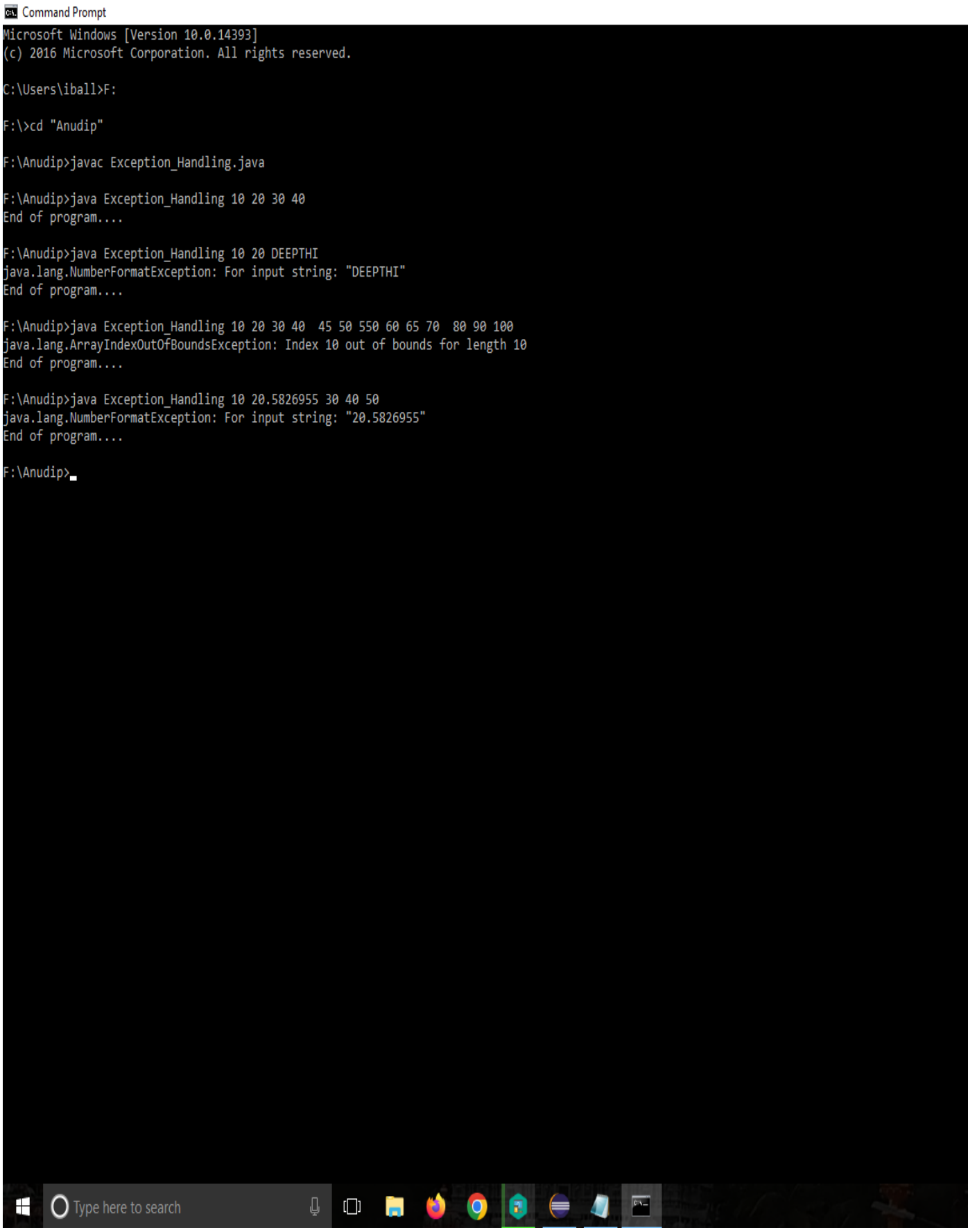
E.V.K.Deepthi (AF0311771)

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

```
class Exception_Handling
{
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....");
}
}
}
```

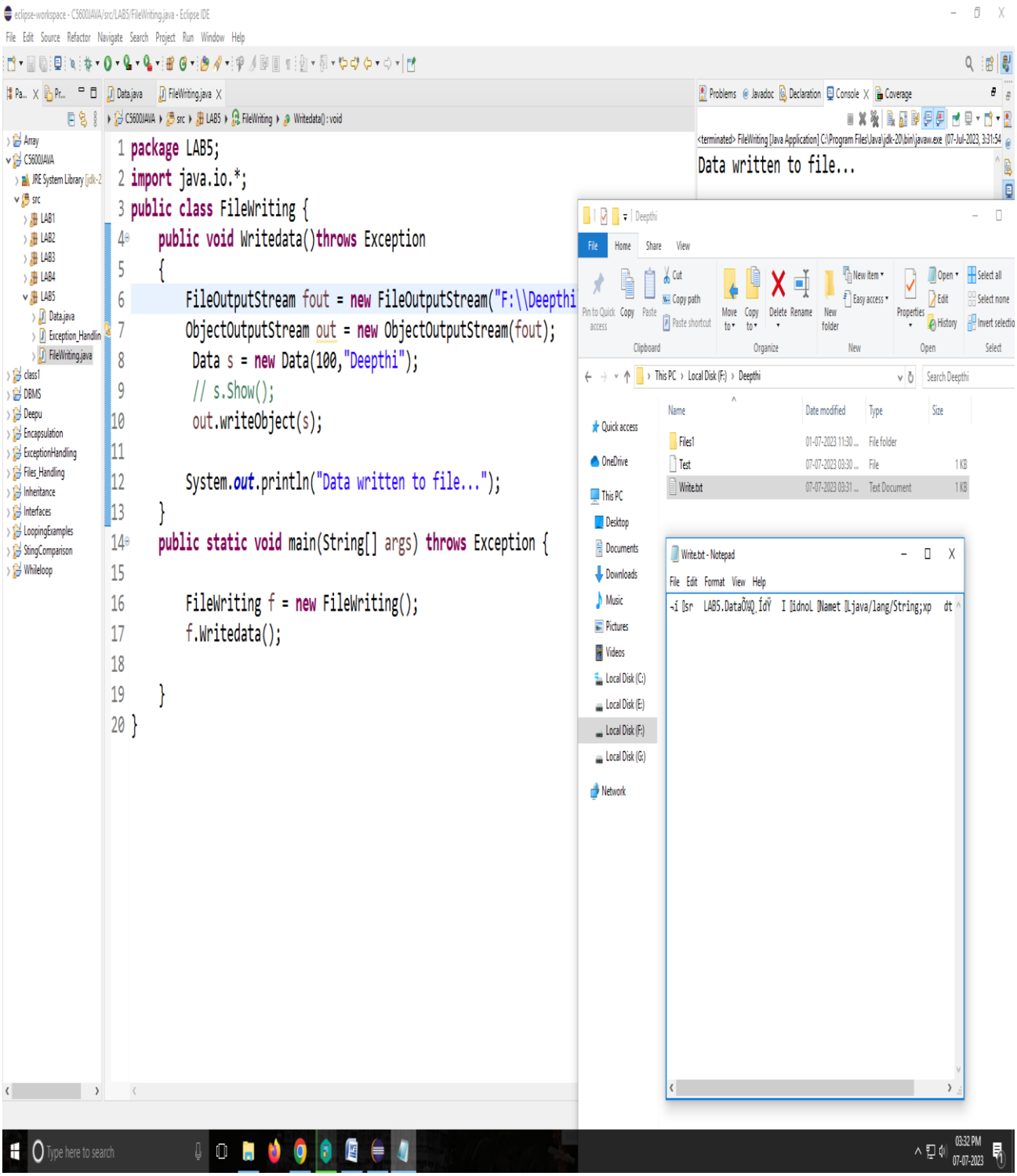
Output:


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>
Type here to search

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

```
package LAB5;
import java.io.*;
public class FileWriting {
	public void Writedata()throws Exception
	{
		FileOutputStream fout = new
FileOutputStream("F:\\Deepthi\\Write.txt");
		ObjectOutputStream out = new ObjectOutputStream(fout);
		 Data s = new Data(100,"Deepthi");
		 // 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();

	}
}
package LAB5;
import java.io.Serializable;
public class Data  implements Serializable {
	int idno;
	String Name;

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

Output:


eclipse-workspace - C5600JAVA/src/LAB5/FileWriting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Data.java
FileWriting.java
1 package LAB5;
2 import java.io.*;
3 public class FileWriting {
4 public void Writedata()throws Exception
5 {
6 FileOutputStream fout = new FileOutputStream("F:\\Deepthi
7 ObjectOutputStream out = new ObjectOutputStream(fout);
8 Data s = new Data(100,"Deepthi");
9 // s.Show();
10 out.writeObject(s);
11
12 System.out.println("Data written to file...");
13 }
14 public static void main(String[] args) throws Exception {
15
16 FileWriting f = new FileWriting();
17 f.Writedata();
18
19 }
20 }
Data written to file...
Deepthi
Files1
Test
Write.txt
Write.txt - Notepad
Type here to search
03:32 PM
07-07-2023

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

```
package LAB5;
import java.util.*;
public class ErrorCondtion {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		int x,y,z;
		try {
			System.out.println("Enter 2 integer values ");
			x=sc.nextInt();
			y=sc.nextInt();

			if(y==0)
			{
				Exception eobj = new Exception("divisor must be non zero value ");
				throw(eobj);

			}
			else
			{
				z=x/y;
				System.out.println("dvivison "+ z);
			}

		}
		catch(Exception e)
		{
			System.out.println(e);

		}
	}
}
```

Output:

Enter 2 integer values
25
5
dvivison 5

(OR)

Enter 2 integer values
9
0
java.lang.Exception: divisor must be non zero value

(OR)

Enter 2 integer values
12
9.6
java.util.InputMismatchException

Q4. Write a java program to illustrate IO exception

```
package LAB5;
import java.io.*;
public class IOExcep {
	public static void main(String[] args) {
		FileReader fileReader = null;
		try {
			// Creating a FileReader object to read a file
			fileReader = new FileReader("input.txt");
			// Reading the contents of the file
			int character;
			while ((character = fileReader.read()) != -1) {
				System.out.print((char) character);
				}
			}
		catch (IOException e)
		{
			// Handling the IOException
			System.out.println("An error occurred while reading the file: " + e);
			}
		finally
		{
			// Closing the FileReader object in the finally block to ensure it's always closed
			try
			{
				if (fileReader != null)
				{
					fileReader.close();
				}
				}
			catch (IOException e)
			{
				System.out.println("An error occurred while closing the file: " + e);
				}
			}
		}
	}
```

Output:

An error occurred while reading the file: java.io.FileNotFoundException: input.txt (The system cannot find the file specified)