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

```
package Exceptionhandling.com;
import java.util.Scanner;
public class Integerarray {
		public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);

			int[] numbers = new int[5];
			try {
			System.out.println("Enter any five integers:");
			for (int i = 0; i < 5; i++) {
			numbers[i] = Integer.parseInt(sc.nextLine());
			}
			} catch (NumberFormatException e) {
		System.out.println("Invalid input! Please enter integers only.");
			} finally {
			sc.close();
			}
			System.out.println("Numbers stored in the array:");
			for (int i = 0; i < 5; i++) {
			System.out.println(numbers[i]);
			}
			}
	}
```

Output:

```
Enter any five integers:
10
20
24
30
```

```
23
Numbers stored in the array:
10
20
24
30
23
```

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

```
package Exceptionhandling.com;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.IOException;
import java.io.Serializable;
    public class Data implements Serializable{
        private String name;
        private int age;
    public Data(String name, int age) {
         this.name = name;
         this.age = age;
  }
     public String getName() {
     return name;
  }
     public int getAge() {
     return age;
}
   public static void main(String[] args)
   {
    Data data = new Data("Apoorva", 10);
    writeDataToFile(data, "d:\\apoorva\\userfile.txt");
    }
   public static void writeDataToFile(Data data, String filename)
```

```
  {
   try (FileOutputStream fileOutputStream = new
FileOutputStream(filename);
ObjectOutputStream objectOutputStream = new
ObjectOutputStream(fileOutputStream))
 {
 objectOutputStream.writeObject(data);
 System.out.println("Data has been written to the file successfully.");
  }   catch (IOException e)
 {
  System.out.println("An error occurred while writing data to the file: "
+ e.getMessage());
  }
  }
  }
```

Output:

¬í sr Exceptionhandling.com.DataÏq¿áb[Ë I ageL namet Ljava/lang/String;xp t Apoorva

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

```
package Exceptionhandling.com;
import java.util.Scanner;
public class Catchexception {
        public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
        System.out.print("Enter a positive integer: ");
        int num = Integer.parseInt(scanner.nextLine());
        if (num <= 0) {
        throw new Exception("Invalid input: Number must be positive");
```

```
        }
        System.out.println("Entered number: " + num);
        } catch (NumberFormatException e) {
        System.out.println("Invalid input: Please enter a valid integer.");
        } catch (Exception e) {
        System.out.println(e.getMessage());
        }
        scanner.close();
        }
}
```

Output:

```
  Enter a positive integer: -12
  Invalid input: Number must be positive
```

4. Write a java program to illustrate IO exception

```
package Exceptionhandling.com;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
   public class IOexception {
        public static void main(String[] args) {
                // TODO Auto-generated method stub
         BufferedReader reader = null;
    try {
    reader = new BufferedReader(new FileReader("file.txt"));
    String line;
  while ((line = reader.readLine()) != null) {
   System.out.println(line);
  }
  } catch (IOException e) {
  System.out.println("An error occurred while reading the file: " +
e.getMessage());
```

```
   e.printStackTrace();
   }
 finally {
 try {
   if (reader != null) {
    reader.close();
  }
  } catch (IOException e) {
  System.out.println("An error occurred while closing the file: " +
e.getMessage());
 e.printStackTrace();
    }
   }
 }
}
```

Output:

```
An error occurred while reading the file: file.txt (The system cannot
find the file specified)
java.io.FileNotFoundException: file.txt (The system cannot find the
file specified)
at java.base/java.io.FileInputStream.open0(Native Method)
at java.base/java.io.FileInputStream.open(FileInputStream.java:219)
at java.base/java.io.FileInputStream.<init>(FileInputStream.java:158)
at java.base/java.io.FileInputStream.<init>(FileInputStream.java:112)
at java.base/java.io.FileReader.<init>(FileReader.java:60)at
Lab5/Exceptionhandling.com.IOexception.main(IOexception.java:13)
```