Week 4.

Aim: To Write a Java program to implement user defined exception handling

Description:

- Java user-defined exception is a custom exception created and throws that exception using a keyword ‘throw’.
- It is done by extending a class ‘Exception’. An exception is a problem that arises during the execution of the program.

Procedure:

- Create a class that extends the Exception class.
- Create a constructor which receives the string as an argument.
- Get the Amount as input from the user.
- If the amount is negative, the exception will be generated.
- Using the exception handling mechanism, the thrown exception is handled by the catch block.
- After the exception is handled, the string “withdraw amount is never negative“ will be displayed.
- If the amount is greater than 0, the message “Please collect the cash “ will be displayed

Program:

```
import java.util.Scanner;

class InvalidAmountException  extends   Exception    // class representing user defined  exception
{
        public    InvalidAmountException(String msg)
        {
                        super(msg);     // calling the constructor of parent Exception
         }
}
// class that uses user defined  exception InvalidAmountException
public    class   DefineUserDefinedException
{
   public static void main(String args[])
   {
        Scanner    s  = new   Scanner(System.in);
        System.out.println(" Enter amount to withdraw ");
        int    amount   =    s.nextInt();
        try
        {
                if(amount<0)     // throw an object of user defined exception
                {
                   throw    new     InvalidAmountException("withdraw amount is never negative");
                }
```

```
                    else
                     {
                                System.out.println(" Please collect cash ");
                     }

          }
         catch (InvalidAmountException ex)
         {
                   System.out.println("Caught the exception");
                     // printing the message from InvalidAmountException object
                   System.out.println("Exception occured: " + ex);
           }

                   System.out.println("rest of the code...");
   }     //main close
}  //class close
```

Output:

```
D:\ACEM\II CSE  Bsection>javac DefineUserDefinedException.java
D:\ACEM\II CSE  Bsection>java DefineUserDefinedException
 Enter amount to withdraw
200
 Please collect cash
rest of the code...
D:\ACEM\II CSE  Bsection>java DefineUserDefinedException
 Enter amount to withdraw
-99
Caught the exception
Exception occured: InvalidAmountException: withdraw amount is never negative
rest of the code...
```

<u>Week 6</u>:

a) **Aim**: To write a java program to split a given text file into n parts. Name each part as the name of theoriginal file followed by .part where n is the sequence number of the part file.

Description:

```
                   BufferedReader    br    =    new    BufferedReader(new FileReader(inputfile));
                   String    strLine;

                   for (int j=1;j<=nof;j++)
                   {
                             FileWriter fw= new FileWriter("File"+j+".txt");      // Destination File Location

                             for (int i=1;i<=nol;i++)
                             {
```

```
                        strLine = br.readLine();

                        if (strLine!= null)
                        {
                                strLine=strLine+"\r\n";
                                fw.write(strLine);
                        }
                }
        }
```

Procedure:

- ➢ Declare variables inputfile as String type nol as double type
- ➢ Take File data type and file as object pass inputfile to File constructor
- ➢ Create Scanner class object
- ➢ Declare count variable and assign to zero
- ➢ Count no. if lines using while loop ,hasNextLine() and nextLine() methods and print no.of lines
- ➢ Find equals parts of file using   double temp = (count/nol);
- ➢ Declare temp1 variable and do typecasting
- ➢ Find split files using BufferedReader

Program:

```
import java.io.*;
import java.util.Scanner;

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

                String    inputfile   =   "test.txt";      //  Source File Name.

                double    nol   =   5.0;   // No. of lines to be split and saved in each output file

                File file = new File(inputfile);

                Scanner scanner = new Scanner(file);

                int count = 0;

                while (scanner.hasNextLine())
                {
                        scanner.nextLine();
                        count++;
                }

                System.out.println("Lines in the file: " + count);      // Displays no. of lines in the input file
                double    temp     =    (count/nol);

                int     temp1    =    (int)temp;
                int      nof      =    0;
                if( temp1  ==  temp)
                {
                        nof  =  temp1;
                }
                else
                {
                        nof  =   temp1+1;
                }
                System.out.println("No. of files to be generated :"+nof);

                // Actual splitting of file into smaller files

                BufferedReader    br    = new    BufferedReader(new   FileReader(inputfile));

                String    strLine;

                for (int j=1;j<=nof;j++)
                {

                        FileWriter    fw    =    new    FileWriter("File"+j+".txt");      // Destination File Location

                        for (int i=1;i<=nol;i++)
                        {
                                strLine   =   br.readLine();

                                if (strLine  !=   null)
                                {
                                        strLine  =   strLine+"\r\n";
                                        fw.write(strLine);
                                }
                        }

                        fw.close();
                }
                    br.close();
                }
                catch (Exception e)
                {
                        System.err.println("Error: " + e.getMessage());
                }
        }
}
```

Output:

```
D:\ACEM\II CSE  Bsection>javac SplitFiles1.java

D:\ACEM\II CSE  Bsection>java SplitFiles1

Lines in the file: 11

No. of files to be generated :3

File1,File2,File3
```

b) Aim: Write a Java program that reads a file name from the user, displays information about whetherthe file exists, whether the file is readable, or writable and the length of the file in bytes.

Description:

A file is a **named location that can be used to store related information**.

For example, main.java is a Java file that contains information about the Java program.

Procedure:

- Create a class FileDemo. Get the file name from the user .
- Use the file functions and display the information about the file.
- getName() displays the name of the file.
- getPath() diplays the path name of the file.
- exists() – Checks whether the file exists or not.
- canRead()-This method is basically a check if the file can be read.
- canWrite()-verifies whether the application can write to the file.
- isDirectory() – displays whether it is a directory or not.
- isFile() – displays whether it is a file or not.
- length()- displays the size of the file.

Program:

```
import java.io.*;
import java.util.*;
class FileDemo
{
public static void main(String args[])
{
        String filename;
        Scanner   s     =   new Scanner(System.in);
        System.out.println("Enter the file name ");
        filename    =   s.nextLine();
        File    f1   =    new File(filename);

        System.out.println("  FILE INFORMATION  ");
        System.out.println("  *****************  ");
        System.out.println(" NAME OF THE FILE  "+f1.getName());
        System.out.println("   PATH OF THE FILE   "+f1.getPath());

        if(f1.exists())
                System.out.println("   THE FILE EXISTS   ");
        else
                System.out.println("   THE FILE DOES NOT EXISTS   ");
```

```
          if(f1.canRead())
                    System.out.println("  THE FILE CAN BE READ ");
          else
                    System.out.println("  THE FILE CANNOT BE READ  ");

          if(f1.canWrite())
                    System.out.println("   WRITE OPERATION IS PERMITTED   ");
          else
                    System.out.println("   WRITE OPERATION IS NOT PERMITTED   ");

          System.out.println("  LENGTH OF THE FILE  "+f1.length()+"  bytes ");

          }
}
```
<u>Output</u>:

```
D:\ACEM\II CSE  Bsection>javac FileDemo.java

D:\ACEM\II CSE  Bsection>java FileDemo

Enter the file name

Test.txt

FILE INFORMATION

*****************

NAME OF THE FILE Test.txt

PATH OF THE FILE Test.txt

THE FILE EXISTS

THE FILE CAN BE READ

WRITE OPERATION IS PERMITTED

LENGTH OF THE FILE  102  bytes
```