Assignment Week-7(Exception Handling)

1. Write a Java program to show the use of all keywords for exception handling.

```
// The program shows exception handling by using a "Sum of Arrays program"
class ExceptionTester {
   public static void main(String[] args) {
      int arr[] = new arr[]{1, 2, 3, 4, 5, 6};
      int sum = 0, i = 0;

      try {
         System.out.println("The array is: {1, 2, 3, 4, 5, 6}");

         while(true){
            sum += arr[i++];
         }
      }

      catch(ArrayIndexOutOfBoundsException e) {
         System.out.println("The Sum is " + sum);
      }

      finally {
         System.out.println("Exiting");
      }
   }
}
```

2.Write a Java program using try and catch to generate Negative Array Index Exception and Arithmetic exception in Java.

```
import java.util.*;
class NegativeArrayIndexException extends Exception {

  public NegativeArrayIndexException(String s) {
    super(s);
  }

}
public class Main
{
        public static void main(String args[])
        {
                        try
                        {
                                Scanner sc=new Scanner(System.in);
                                int arr[]={1,2,3,4,5};
                                System.out.println("Enter the index of the array which you want to
retrieve");
                                int index=sc.nextInt();
                                if(index<0)
                                throw NegativeArrayIndexException("Index less than 0");
                                else
                                System.out.println(arr[index]);
                                System.out.println("Enter divisor and dividend for a division");
                                int divisor=sc.nextInt();
                                int dividend=sc.nextInt();
                                if(divisor==0)
                                throw ArithmeticException;

                        }
                        catch(NegativeArrayIndexExceptionobj)
                        {
                                System.err.println(obj);
                        }
                        catch(ArithmeticException obj)
                        {
                                System.out.println("Caught Arithmetic exception!");
                        }

        }
```

3.Define an exception called “NoMatchFoundException” that is thrown when a string is not equal to “University”. Write a program that uses this exception.

```
import java.io.*;
class NoMatchFoundException extends Exception {
  static int count=0;
  public NoMatchFoundException(String s) {
    super(s);
  }
  public static void countString(String s) throws NoMatchFoundException {
    if  (s.compareTo("University")==0) {
      count++;
      System.out.println("Count of University used="+count);
    }
    else {
      throw new NoMatchFoundException(s + " is not University");
    }


  }

}
public class Main
{
	public static void main(String args[])throws IOException
	{
		BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
		System.out.println("Enter strings one by one,Enter E to exit");
		while(true)
		{
			String c=br.readLine();
			if(c.compareTo("E")==0)
			break;
			try
			{
				NoMatchFoundException ob=new
NoMatchFoundException("Test");
				ob.countString(c);
			}
			catch(NoMatchFoundException obj)
			{
				System.err.println(obj);
```

```
                        }
                }
        }
}
```

4. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.

```
 import java.io.*;
class NotALetterException extends Exception {

  static int count=0;

  public NotALetterException(String s) {
    super(s);
  }


  public static void countLetter(char c) throws NotALetterException {


    if  ((c>=65&&c<=90)||(c>=97&&c<=122)) {
      count++;
      System.out.println("Count of characters="+count);
    }
    else {
      throw new NotALetterException(c + " is not a letter");
    }


  }

}
public class Main
{
        public static void main(String args[])throws IOException
        {
                BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter characters one by one,Enter E to exit");
                while(true)
                {
                        char c=br.readLine().charAt(0);
                        if(c=='E')
                        break;
                        try
                        {
                                NotALetterException ob=new NotALetterException("Test");
                                ob.countLetter(c);
                        }
                        catch(NotALetterException obj)
                        {
                                System.err.println(obj);
                        }
                }
        }
}
```

5. WAP called Factorial.java that computes factorials and catches the result in an array of type long for reuse. The long type of variable has its own range. For example 20! Is as high as the range of long type . So check the argument passed and throw an exception if it is too big or too small

```
import java.io.*;
class IllegalArgumentException extends Exception
{
  public IllegalArgumentException(String x)
  {
    super(x);
  }
}
class fact extends Factorial{
  arr=[];
  super(arr,x);
}
class Main{
  public static void main(String args[]){
    arr=[];
    arr = fact(arr,x); //calls the Factorial.java
    int x = Integer.parseInt(br.readLine());
    try
    {
      if(x<0)
      {
        throw new IllegalArgumentException("Value of x must be positive");
      }
      if(x>arr.length){
        throw new IllegalArgumentException("Result will overflow")
      }
    }
  }
}
```

6. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.

```
 import java.io.*;
class NotALetterException extends Exception {

  static int count=0;

  public NotALetterException(String s) {
    super(s);
  }


  public static void countLetter(char c) throws NotALetterException {


    if  ((c>=65&&c<=90)||(c>=97&&c<=122)) {
      count++;
      System.out.println("Count of characters="+count);
    }
    else {
      throw new NotALetterException(c + " is not a letter");
    }



  }

}
public class Main
{
        public static void main(String args[])throws IOException
        {
                BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter characters one by one,Enter E to exit");
                while(true)
                {
```

```
                                char c=br.readLine().charAt(0);
                                if(c=='E')
                                break;
                                try
                                {
                                        NotALetterException ob=new NotALetterException("Test");
                                        ob.countLetter(c);
                                }
                                catch(NotALetterException obj)
                                {
                                        System.err.println(obj);
                                }
                        }
                }
        }
```

7. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a “NoMatchFoundException” when it fails to print the capital of the country entered at the command line.

```
 import java.io.*;
class NoMatchFoundException extends Exception {

    public NoMatchFoundException(String s) {
    super(s);
  }


  public static void printcapital(String S) throws NoMatchFoundException {


    if  (S.equals("India")) {
      System.out.println("New Delhi");
    }
    else {
      throw new NoMatchFoundException( " Failed to find Capital");
    }


  }

}
public class Main
```

```
{
	public static void main(String args[])
	{

			try
			{
				NoMatchFoundException ob=new
NoMatchFoundException("Test");
				ob.printcapital(args[0]);
			}
			catch(NoMatchFoundException obj)
			{
				System.err.println(obj);

			}

	}
}
```

8. Write a program that takes a value at the command line for which factorial is to be computed. The program must convert the string to its integer equivalent. There are three possible user input errors that can prevent the program from executing normally.
The first error is when the user provides no argument while executing the program and an ArrayIndexOutOfBoundsException is raised. You must write a catch block for this.
The second error is NumberFormatException that is raised in case the user provides a non-integer (float double) value at the command line.
The third error is IllegalArgumentException. This needs to be thrown manually if the value at the command line is 0.

```
class IllegalArgumentException extends Exception {

  public IllegalArgumentException(String s) {
    super(s);
  }

}
public class Main
{
	public static void main(String args[])
	{
			try
```

```
                    {
                              if(Integer.valueOf(args[0])==0)
                              throw new IllegalArgumentException("Argument entered 0");
                              if(args[0].contains("."))
                              throw new NumberFormatException();

                    }
                    catch(ArrayIndexOutOfBoundsException obj)
                    {
                              System.out.println("Caught Array Index out of bounds exception");
                    }
                    catch(NumberFormatException obj)
                    {
                              System.out.println("Caught Number Format exception");
                    }
                    catch(IllegalArgumentException obj)
                    {
                              System.err.println(obj);
                    }
          }
}
```

9. Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of argument is less than 5, throw the CheckArgumentexception,else print the addition of all the five numbers.

```
import java.util.Scanner;
import java.io.*;
class CheckArgumentexception extends Exception
{
CheckArgumentexception(String message)
{super(message);
}
}
public class Main
{
public static void main(String args[])
{

int a,i,count=0,sum=0,avg;
for( i=0;i<args.length;i++)
{ a=Integer.parseInt(args[i]);
count=count+1;
sum=sum+a;
}
avg=sum/i;
try
{
if(count<5)
{
throw new CheckArgumentexception("Arguments are less than 5");
}
 else
{
   System.out.println(sum);
   System.out.println(avg);
}
 }
catch (CheckArgumentexception e)
{System.err.println(e);
}


}
}
```

10. Consider a Student examination database system that prints the mark sheet of students. Input the following from the command line.
(a) Student's Name
(b) Marks in six subjects
These marks should be between 0 to 50. If the marks are not in the specified range, raise a RangeException, else find the total marks and prints the percentage of the students.

```
import java.util.Scanner;
import java.io.*;
class RangeException extends Exception
{
        RangeException(String message)
        {
                super(message);
        }
}
public class Main
{
        public static void main(String args[])
        {
```

```
                    int a,i,total=0;
                    float percentage;
                    for( i=1;i<args.length;i++)
                    {
                              a=Integer.parseInt(args[i]);
                              try{
                                        if(a<0||a>50)
                                        throw new RangeException("Marks not between 0 and 50");
                                        else{
                                                  total=total+a;
                                                  percentage=(float)total/(float)i;
                                                  if(i==args.length-1)
                                                  System.out.println("Total
Marks="+total+"Percentage="+percentage);
                                              }
                                                   }
                              catch (RangeException e)
                              {
                                        System.err.println(e);
                              }

                    }

          }

}
```

21. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative.

```
import java.util.Scanner;

class AgeNegativeException extends Exception {

 public AgeNegativeException(String msg) {

  super(msg);
 }
}

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

  Scanner s = new Scanner(System.in);
  System.out.print("Enter ur name :: ");
  String name = s.nextLine();
  System.out.print("Enter ur age :: ");
  int age = s.nextInt();

  try {
   if(age < 0)
    throw new AgeNegativeException("Age must be greater than 0");
   else
    System.out.println("Valid age !!!");
  }
  catch (AgeNegativeException a) {
   System.out.println("Caught an exception");
   System.out.println(a.getMessage());
  }
 }
}
```

22. Write a program to throw user defined exception if the given number is not positive.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class MyException extends Exception
{
 public MyException(String str)
 {
  System.out.println(str);
 }
}

public class SignException {

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

  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  System.out.print("Input number :: ");

  try {
   int num = Integer.parseInt(br.readLine());
   if(num < 0)
    throw new MyException("Number is negative");
   else
    throw new MyException("Number is positive");
  }
  catch (MyException m) {
   System.out.println(m);
  }
 }

}
```

23. Write a program to throw a user-defined exception "String Mismatch Exception", if two strings are not equal (ignore the case).

```
import java.util.Scanner;

class StringMismatchException extends Exception {

 public StringMismatchException(String str) {

  System.out.println(str);
 }
}
public class StringExcDemo {

 public static void main(String[] args) {

  Scanner scan = new Scanner(System.in);
  System.out.print("Enter the string :: ");
  String input = scan.nextLine();

  try {
   if(input.equalsIgnoreCase("Hello"))
    System.out.println("String matched !!!");
   else
    throw new StringMismatchException("String not matched ???");
  }
  catch (StringMismatchException s) {
   System.out.println(s);
  }
 }

}
```

24. Design a stack class. Provide your own stack exceptions namely push exception and pop exception, which throw exceptions when the stack is full and when the stack is empty respectively. Show the usage of these exceptions in handling a stack object in the main.

```
public class ArrayStack implements Stack
  {
    public int[] item;
    public int stackTop;

    public ArrayStack( int size )
    {
         item = new int[size];      // Make array
         stackTop = 0;
    }

    public void push( int x ) throws StackException
    {
         if ( stackTop == item.length )
         {
           throw new StackException("Stack overflow");
         }

         item[stackTop] = x;         // Store x in next slot
         stackTop++;                 // Advance one slot location
    }

    public int  pop ( ) throws StackException
    {
         int returnItem;

         if ( stackTop == 0 )
         {
           throw new StackException("Stack underflow");
         }

         returnItem = item[ stackTop-1 ];   // Get last stored item
         stackTop--;                        // Back up one slot location

         return returnItem;
    }
  }
   public static void main( String[] args )
  {
    int   x;
    Stack s;

    s = new ArrayStack( 6 );    // Will cause underflow

    try
    {
      x = 4; s.push(x); System.out.println("push(" + x + ");");
      x = 7; s.push(x); System.out.println("push(" + x + ");");
      x = 8; s.push(x); System.out.println("push(" + x + ");");
      x = 9; s.push(x); System.out.println("push(" + x + ");");

      x = s.pop(); System.out.println("pop() ---> " + x );
      x = s.pop(); System.out.println("pop() ---> " + x );
      x = s.pop(); System.out.println("pop() ---> " + x );
      x = s.pop(); System.out.println("pop() ---> " + x );
      x = s.pop(); System.out.println("pop() ---> " + x );
    }
    catch ( StackException e )
    {
      System.out.println("Error detected: " + e.getMessage() );
      System.exit(1);
    }
  }
```

25. Write an application that displays a series of at least five student ID numbers (that you have stored in an array) and asks the user to enter a numeric test score for the student. Create a ScoreException class, and throw a ScoreException for the class if the user does not enter a valid score (zero to 100). Catch the ScoreException and then display an appropriate message. In addition, store a 0 for the student's score. At the end of the application, display all the student IDs and scores.

```
import java.util.*;
class Student {
String id;
int score;
public Student (String id) {
this.id = id;
score = 0;
}
}
class ScoreException extends Exception {
ScoreException (String errorMessage) {
```

```
super(errorMessage);
}
}
public class ScoreTest {
public static void main (String [] args) {
Scanner sc = new Scanner(System.in);
Student students[] = new Student[10];
for (int i = 0; i < 10; ++i) {
String randomUUID =
UUID.randomUUID().toString().replaceAll("-", "").substring(0, 10);
// Random String for the glory of Satan...
students[i] = new Student(randomUUID); // Create a new instance
System.out.print("\n\tStudent " + (i+1) + ", id: " +
students[i].id + "\n\tEnter score (0-100): ");
int score = 0;
try {
score = sc.nextInt(); // Input the score
if (score < 0 || score > 100) {
score = 0;
throw new ScoreException("Invalid Score. Using
default 0 value."); // Throw the exception and set value to 0
}
}
catch (ScoreException exception) {
System.err.println("\n\tError: " + exception);
}
finally {
students[i].score = score;
// Finally set the score value to the inputted value or 0
}
}
System.out.println("\n\tID\t\tScore");
for (Student stud : students)
System.out.println("\t" + stud.id + "\t" + stud.score);
}
}
```