Assignment-6

Name: Sandeep Kumar Singh

Reg. No: 2021CA094

Q1. Write a Java program to demonstrate

(i) Arithmetic Exception

Code</>

```
public class ArithmeticException
{
void divide(int a, int b)
{
// performing divison and storing th result
int res = a / b;
System.out.println("Division process has been done successfully.");
System.out.println("Result came after division is: " + res);
}
// main method
public static void main(String argvs[])
{
// creating an object of the class ArithmeticException
ArithmeticException obj = new ArithmeticException();
obj.divide(1, 0);
}
```

```
}
```

(ii) NullPointer Exception

Code</>

```
// To use randomUUID function.
import java.util.UUID;
import java.io.*;
class Singleton
{
// Initializing values of single and ID to null.
private static Singleton single = null;
private String ID = null;
private Singleton()
{
/* Make it private, in order to prevent the
creation of new instances of the Singleton
class. */
// Create a random ID
ID = UUID.randomUUID().toString();
}
public static Singleton getInstance()
{
if (single == null)
single = new Singleton();
return single;
```

```
}
public String getID()
{
return this.ID;
}
}
// Driver Code
public class TestSingleton
{
public static void main(String[] args)
{
Singleton s = Singleton.getInstance();
System.out.println(s.getID());
}
}
```

(iii) StringIndexOutOfBound Exception

code</>

```
// Java program to demonstrate the
// ArrayIndexOutOfBoundException
// import the required package
import java.io.*;
import java.lang.*;
import java.util.*;
// driver class
```

```
class MainClass {
// main method
public static void main(String[] args)
{
// declaring and initializing an array of length 4
int[] x = { 1, 2, 3, 4 };
// accessing the element at 0 index
System.out.println(x[0]);
// accessing an index which is greater than the
// length of array
System.out.println(x[10]);
// accessing a negative index
System.out.println(x[-1]);
}
}
```

(iv) NumberFormat Exception

code</>

```
// Java Program to Demonstrate Working of parseInt() Method
// Where NumberFormatException is Thrown
// Main class
public class Exceptionclass {
// Main driver method
public static void main(String args[])
{
```

```
// Custom wide-varied inputs to illustrate
// usage of valueOf() method
int decimalExample = Integer.parseInt("20");
int signedPositiveExample = Integer.parseInt("+20");
int signedNegativeExample = Integer.parseInt("-20");
int radixExample = Integer.parseInt("20", 16);
int stringExample = Integer.parseInt("geeks", 29);
// It will raise NumberFormatException
String invalidArguments = "";
int emptyString
= Integer.parseInt(invalidArguments);
int outOfRangeOfInteger
= Integer.parseInt("Helloworld", 29);
int domainOfNumberSystem
= Integer.parseInt("Hello", 28);
// Print commands on console
System.out.println(decimalExample);
System.out.println(signedPositiveExample);
System.out.println(signedNegativeExample);
System.out.println(radixExample);
System.out.println(stringExample);
}
}
```

(v) User-Defined Exceptions

code</>

```
// A Class that represents use-defined exception
class MyException extends Exception {
public MyException(String s)
{
// Call constructor of parent Exception
super(s);
}
}
// A Class that uses above MyException
public class Main {
// Driver Program
public static void main(String args[])
{
try {
// Throw an object of user defined exception
throw new MyException("HelloWorld");
}
catch (MyException ex) {
System.out.println("Caught");
// Print the message from MyException object
System.out.println(ex.getMessage());
}
}
```

```
}
```

Q2. What is the output of the following codes?

(i)

```
public class JavaHungry {
public static void main(String args[])
{
try
{
System.out.print("A");
int num = 99/0;
System.out.print("B");
}
catch(ArithmeticException ex)
{
System.out.print("C");
}
catch(Exception ex)
{
System.out.print("D");
}
System.out.print("E");
}
}
```

Output-ACE

(ii). public class JavaHungry {

```
public static void main(String args[])
{
try
{
System.out.print("A");
int num = 99/0;
System.out.print("B");
}
catch(ArithmeticException ex)
{
System.out.print("C");
}
catch(Exception ex)
{
System.out.print("D");
}
finally
{
System.out.print("E");
}
}
}
```

OUTPUT- ACE

(iii). public class JavaHungry {

public static void main(String args[])

{

try

{

System.out.print("A");

int num = 99/0;

System.out.print("B");

}

System.out.print("C");

catch(ArithmeticException ex)

{

System.out.print("D");

}

}

}

Output- ERRORMain.java:7:error:'try' without 'catch', 'finally' of resource declrations

try

^

Maint.java:21 error: 'catch' without 'try'

catch(ArithmeticException ex)

(iv). public class JavaHungry {

public static void main(String args[])

{

```
try
{
System.out.print("A");
int num = 99/0;
System.out.print("B");
}
catch(ArithmeticException ex)
{
System.out.print("C");
System.exit(0);
}
catch(Exception ex)
{
System.out.print("D");
}
finally
{
System.out.print("E");
}
System.out.print("F");
}
}
```

OUTPUT - AC