Lab Assignment-09

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 14/04/22

QUES 1: Write a Java program to generate an ArrayIndexOutofBoundsException and handle it using catch statement.

SOLUTION:

```
class Lab9Q1 {
    public static void main(String args[]) {
        int array[] = new int[10];
        try {
            array[20] = 10;
        } catch (Exception e) {
            System.out.println("Array Index Out Of Bounds Exception!!!");
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
Array Index Out Of Bounds Exception!!!
java.lang.ArrayIndexOutOfBoundsException: Index 20 out of bounds for length 10
```

QUES 2: A subclass exception must appear before super-class exception. Justify this with suitable Java programs.

SOLUTION:

```
public class Lab9Q2 {
    public static void main(String[] args) {
        try {
            int result = 5 / 0;
            System.out.println("Result is: " + result);
        } catch (ArithmeticException e2) {
            System.out.println("Subclass ArithmeticException Occured");
        } catch (Exception e1) {
            System.out.println("Superclass Exception Occured");
        }
        System.out.println("Subclass Exception Appeared First");
    }
}
```

OUTPUT 1:

```
Subclass ArithmeticException Occured
Subclass Exception Appeared First
```

SOLUTION:

```java
public class Lab9Q2 {
    public static void main(String[] args) {
        try {
            int result = 5 / 0;
            System.out.println("Result is: " + result);
        } catch (Exception e1) {
            System.out.println("Superclass Exception Occured");
        } catch (ArithmeticException e2) {
            System.out.println("Subclass ArithmeticException Occured");
        }
        System.out.println("Block of code becomes unreachable");
    }
}
```

OUTPUT 2:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
        Unreachable catch block for ArithmeticException. It is already handled by the catch
block for Exception

        at Lab9Q2.main(Lab9Q2.java:9)
```

QUES 3: Write a Java program to illustrate try..catch..finally block.

SOLUTION:

```java
class Lab9Q3 {
    public static void main(String args[]) {
        int array[] = new int[10];
        try {
            array[20] = 10;
        } catch (Exception e) {
            System.out.println("Catch Block Executed!!!");
            System.out.println("Array Index Out Of Bounds Exception!!!");
            System.out.println(e);
        } finally {
            System.out.println("Finally Block Executed!!!");
        }
    }
}
```

OUTPUT:

```
Catch Block Executed!!!
Array Index Out Of Bounds Exception!!!
java.lang.ArrayIndexOutOfBoundsException: Index 20 out of bounds for length 10
Finally Block Executed!!!
```

QUES 4: Write a Java class which has a method called ProcessInput(). This method checks the number entered by the user. If the entered number is negative then throw an user defined

exception called NegativeNumberException, otherwise it displays the double value of the entered number.

SOLUTION:

```
class NegativeNumberException extends Exception {
    public NegativeNumberException(String str) {
        System.out.println(str);
    }
}

class Check {
    public void ProcessInput(int num) {
        try {
            if (num < 0) {
                throw new NegativeNumberException("Number is negative");
            } else {
                System.out.println("Double Number: " + (2 * num));
            }
        } catch (Exception e) {
            System.out.println("Caught Exception Negative Number!!!");
        }
    }
}

class Lab9Q4 {
    public static void main(String args[]) {
        int a = -5;
        Check obj = new Check();
        obj.ProcessInput(a);
    }
}
```

OUTPUT 1:

```
Number is negative
Caught Exception Negative Number!!!
```

OUTPUT 2:

```
Double Number: 10
```

QUES 5: Write a program to create user defined exceptions called HrsException, MinException and SecException. Create a class Time which contains data members hours, minutes, seconds and throw the user defined exceptions if hours (>24 & <0), minutes (>24 & <0), seconds (>60 & <0).

SOLUTION:

```
class HrsException extends Exception {
    public HrsException(String str) {
```

```
        System.out.println(str);
    }
}

class MinException extends Exception {
    public MinException(String str) {
        System.out.println(str);
    }
}

class SecException extends Exception {
    public SecException(String str) {
        System.out.println(str);
    }
}

class Time {
    public static void main(String args[]) {
        int hours, minutes, seconds;
        hours = 12;
        minutes = 25;
        seconds = 30;
        try {
            if (hours < 0 || hours > 24) {
                throw new HrsException("Hours Exception Caught!!!");
            } else if (minutes < 0 || minutes > 60) {
                throw new MinException("Minutes Exception Caught!!!");
            } else if (seconds < 0 || seconds > 60) {
                throw new MinException("Minutes Exception Caught!!!");
            } else {
                System.out.println("Time: " + hours + " hrs : " + minutes + " mins: " +
seconds + " seconds");
            }
        } catch (Exception e) {
            System.out.println("Exception Caught!!!");
        }
    }
}
```

OUTPUT:

```
Minutes Exception Caught!!!
Exception Caught!!!
```
