COMP 6481

Tutorial 3:

Exception Handling + Review

What is Exception

- Condition that has occurred in a piece of code
- Java provides a handling mechanism for Exception by creating different types of objects which describes the cause of exception
- Exception comes as an object
- All the Exceptions are instances of Throwable or it’s classes down the hierarchy
- New Exception class can be created by extending Exception

Exception handling

- Try, catch, throw, throws, finally
- Exception raised in try, can be caught by catch block.
- finally will always be executed

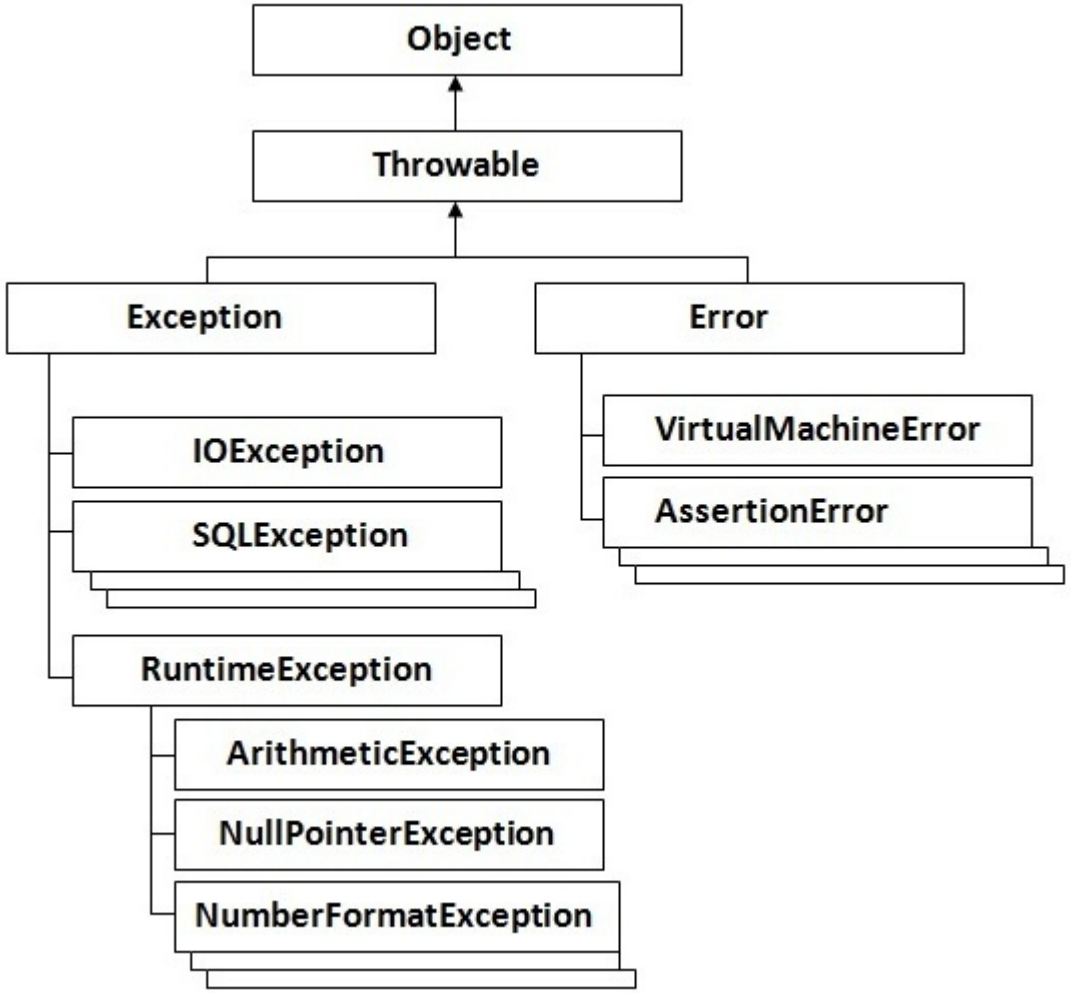
```
try{
        //code
}catch(ExceptionType e1) {
        //Flow 1
}catch(ExceptionType e2){
        //Flow 2
} finally{
//close file
//close database connection
//close network socket
}
```

Powerful catch

- Exception or any such parent catch block allows Exception to be caught of any child type as well.

```
try{
        //code
}catch(Exception e1) {
        //Flow 1
}
```

Exception hierarchy



Types

- Checked – Checked at compile time. The classes that extends the Throwable except RuntimeException and Error.
- Unchecked – Checked at runtime. Classes extending RuntimeExceptions.
- Error – Can not be recovered. Eg. OOM

Question 1

Match each situation in the first column with an item in the second column.

1. int[] A;
   A[0] = 0;

2. The Java VM starts running your program, but the VM can't find the Java platform classes. (The Java platform classes reside in classes.zip or rt.jar.)

3. A program is reading a stream and reaches the end of stream marker.

4. Before closing the stream and after reaching the end of stream marker, a program tries to read the stream again.

1.Error

2.checked exception

3. Compile Error

4.no exception

Question 2

Match each situation in the first column with an item in the second column.

1. int a = 30, b =0;
   int c = a/b;

2. String a =null;
   System.out.print(a.charAt(0));

3. int num = Integer.parseInt("XYZ");
   System.out.print(num);

4. int array[] = new int[5];
   array[6] = 9;

1. NullPointerException

2. ArrayOutOfBoundsException

3. ArithmeticException

4. NumberFormatException

Question 3

Modify the following cat method so that it will compile:

```
public static void cat(File named) {
    RandomAccessFile input = null;
    String line = null;
    try {
        input = new RandomAccessFile(named, "r");
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        return;
    } finally {
        if (input != null) {
            input.close();
        }
    }
}
```

Question 4

Modify the following method so that it will compile:

```
package data;

import java.io.File;
import java.io.IOException;
import java.sql.SQLException;

public class BadIO {
	public static void cat(File named) {
		BadIO obj_IO = new BadIO();

		try{
			obj_IO.fileBlowUp();
			obj_IO.databaseBlowUp();
		} //INSERT CODE HERE
	}

	void databaseBlowUp() throws SQLException {
		throw new SQLException();
	}

	void fileBlowUp() throws IOException {
		throw new IOException();
	}
}
```

Question 5

What is the output of the following program?

```
class Base {
        public void print() {
                System.out.println("Base");
        }
}
class Derived extends Base {
        public void print() {
                System.out.println("Derived");
        }
}
class Main{
        public static void doPrint( Base o ) {
                o.print();
        }
        public static void main(String[] args) {
                Base x = new Base();
                Base y = new Derived();
                Derived z = new Derived();
                doPrint(x);
                doPrint(y);
                doPrint(z);
        }
}
```

Question 6

What are the errors in the following program and how they can be fixed ?

```
public class A {
    private int a = 100;
    public void setA( int value) {
        a = value;
}    public int getA() {
        return a;
    }
}
```

```
public class OOPExercises {
    public static void main(String[] args) {
        A objA = new A();
        System.out.println("in main(): ");
        System.out.println("objA.a = "+objA.a);
        objA.a = 222;
    }
}
```