Week 12-2 : Exception Handling

Part 3. Exception Handling
To handle a run-time error.

- try-catch-finally

```java
try {
    // Execute statement : error occur possibly
}
catch (Declaring Exception type for handling) {
    // Exception handling statement
}
finally {
    // Execute unconditionally. Possibly being omitted.
}
```

- kinds of an exception

<table>
<thead>
<tr>
<th>kinds of an exception</th>
<th>case of an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArithmeticException</td>
<td>Time when an integer is divided by 0</td>
</tr>
<tr>
<td>NullPointerException</td>
<td>Time when a Null reference is refered.</td>
</tr>
<tr>
<td>ClassCastException</td>
<td>Time when a object is changed to an unavailable type.</td>
</tr>
<tr>
<td>OutOfMemoryException</td>
<td>Time when memory is a lack.</td>
</tr>
<tr>
<td>ArrayIndexOutOfBoundsException</td>
<td>Time when an array is accessed out of index</td>
</tr>
<tr>
<td>IllegalArgumentException</td>
<td>Pass a wrong parameter</td>
</tr>
<tr>
<td>IOException</td>
<td>I/O fail or interrupt occur</td>
</tr>
<tr>
<td>NumberFormatException</td>
<td>Time when a string representing number is changed to not accordant type of the numbers</td>
</tr>
</tbody>
</table>
- Exception about accessing array out of index

```java
int intArray[] = new int[4];
try {
    intArray[3] = 15;
    intArray[5] = 20;
}
catch(ArrayIndexOutOfBoundsException e) {
    System.out.println("out of index");
}
```

- Making my own exception

```java
class MyException extends Exception {
    public MyException() {
    }

    public MyException(String msg) {
        super(msg);
    }
}

class MyException:
    def __init__(self):
        super().__init__()

    def __init__(self, msg):
        super.__init__(msg)

class FullConstructors {
    public static void f() throws MyException {
        System.out.println("Throwing MyException from f()");
        throw new MyException();
    }

    public static void g() throws MyException {
        System.out.println("Throwing MyException from g()");
        throw new MyException("Originated in g()");
    }

    public static void main(String[] args) {
        try {
            f();
        } catch (MyException e) {
            e.printStackTrace(System.out);
        }
        try {
            g();
        }
```


```java
} catch (MyException e) {
    e.printStackTrace(System.out);
} }
```

If error occur at each functions using throw, the error was caught at the catch, and the error was handled in a MyException class.

[Exercise]
1. Make a division program. Using try-catch, handle a exception about divided by 0 using try-catch.

2. Make a changing type program. Using try-catch, handle exceptions about several cases.
   
   ```java
   string[] stringNum = {"1", "10", "1.11"};
   ```

   -hint
   Integer.parseInt

Distributed Computing Systems Lab
CSE, SNU