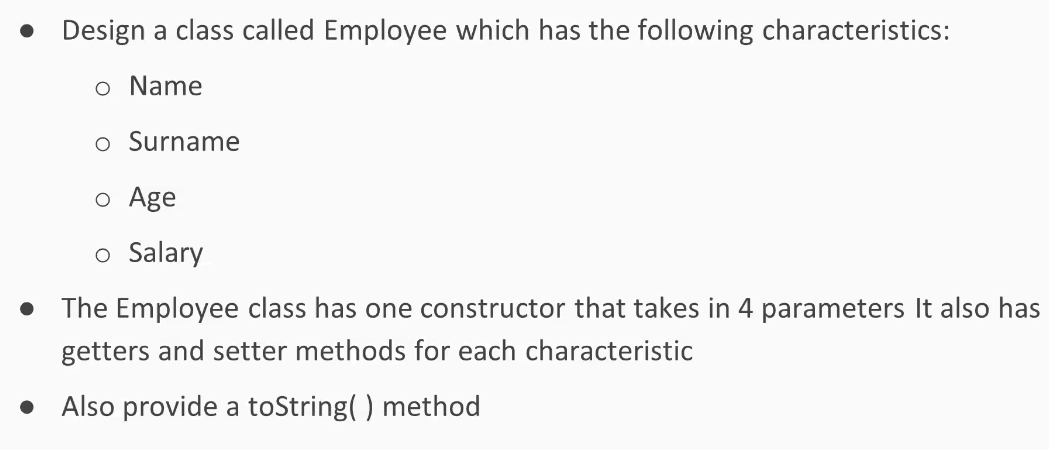
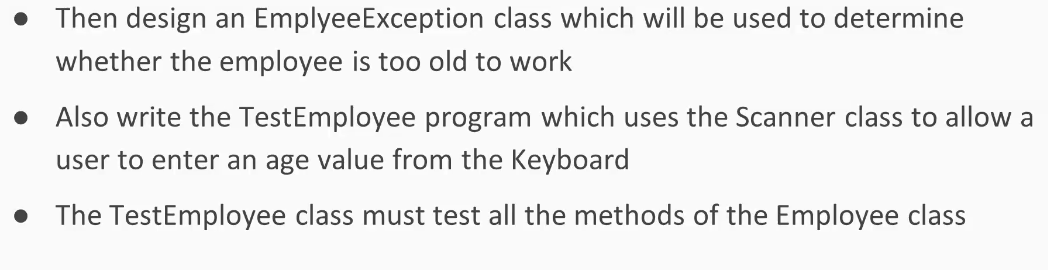
Hands On Exercises - Exception Handling

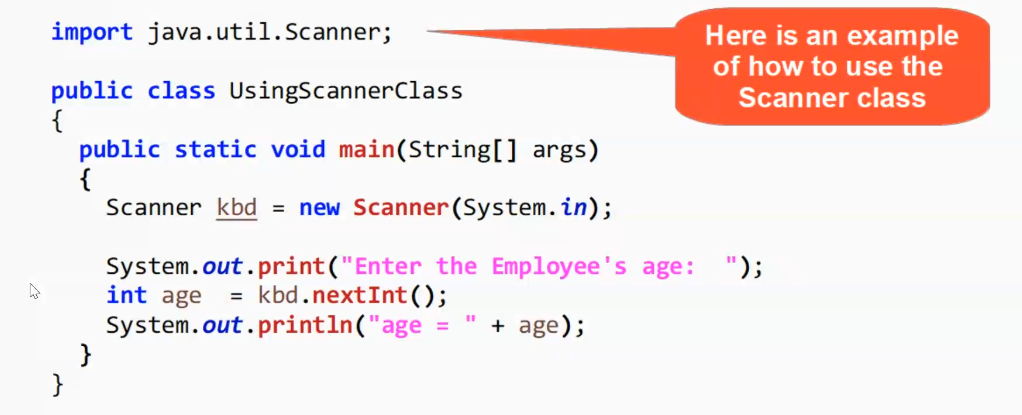
**Setup Instructions:**

1. Create a new Java Project.
2. Create your classes in the package “exceptions”
3. Define your classes as given below for each assignment.

Assignment 01



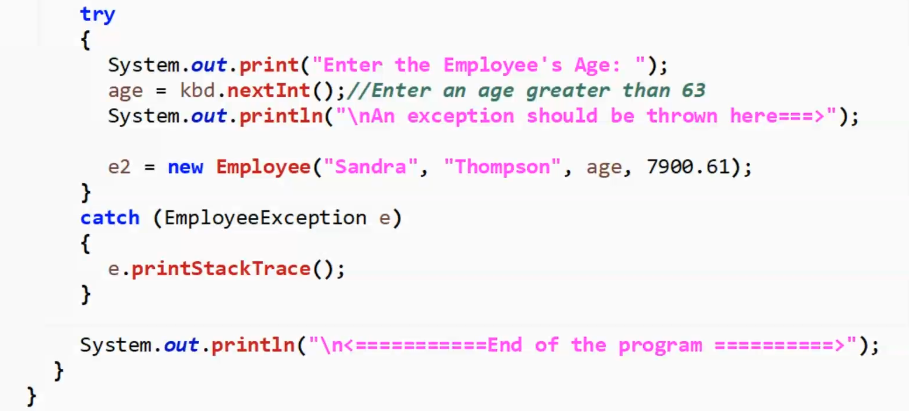
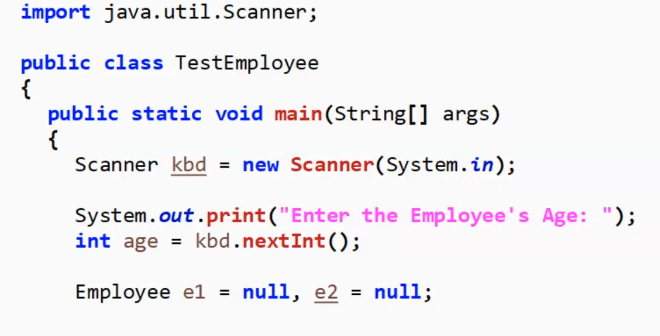




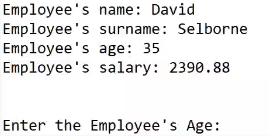
Instructions:

1. The Employee class constructor must check to see if the age>63.
2. If the age>63, throw a new EmployeeException passing in the String “Age cannot be older than 63.”
3. Set the Employee class instance variables equal to the arguments passed in to the constructor.
4. In the setter for the age variable, also check if the age>63 and if so, throw a new EmployeeException passing in the String “Age cannot be older than 63.”

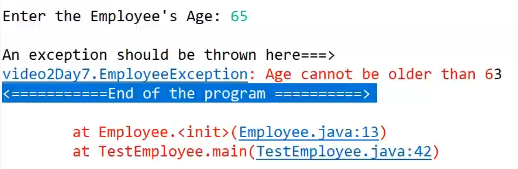
Use the below code to set up your TestEmployee class:



Run your TestEmployee program to test both the happy :) path



and the exception scenario:



Assignment 02

##### **Handling Stuff**

 Coding

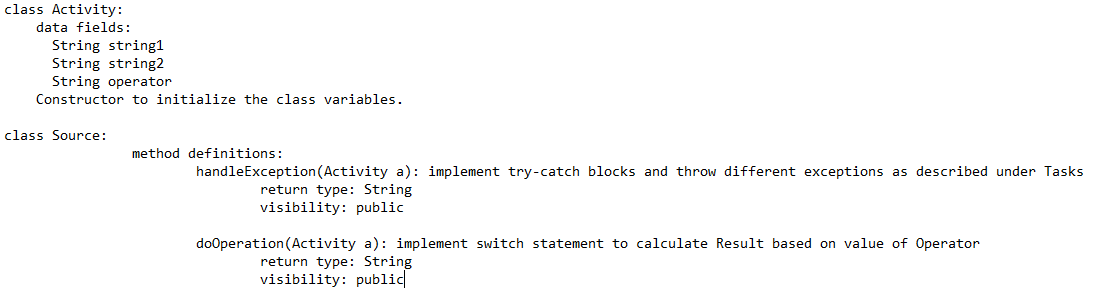
##### Description

In **Java**, we can use more than one catch block with the try block. Generally, multiple catch block is used to handle different types of exceptions, which means each catch block is used to handle different types of exceptions.

If you use multiple catch blocks for the same type of exception, then it will give you a compile-time error because **Java does not allow you to use multiple catch block for the same type of exception**. A catch block is always preceded by the try block.

Write a program to demonstrate Multiple Exceptions.

**Specifications:**



You have to implement the following methods under Source class:

* **handleException (Activity a)** - In this function you have to check for exceptions.
* **doOperation (Activity a)** - this function should implement the string operation between **string1** and **string2** for the operator **operator**.
* If **operator = '+'**, concat the strings **string1** and **string2**.
* **e.g.** for **string1 = "hello"**and **string2 = "world"**, then **result** = **"helloworld"**
* If **operator = '-'**, replace the contents of **string2** in **string1** with empty string.
* **e.g.** If **string1 = "helloworld"** and **string2 = "world"**, then **result = "hello"**

**Tasks:**

In the function **handleException** **(Activity a)**:

* Check that the value of either **string1** or **string2** variable is **null**, then throw appropriate exception for **NullPointerException** and return "**Null values found**".
* Check if the value of **operator** variable is not equal to these string operators ((+ or -) using logical AND operator. If the condition is true then throw and return the default exception with the Operator as the return message.
* If no exception is found return "**No Exception Found**".

In the function **doOperation (Activity a)**:

* perform the string operations, using switch statement and return the correct value.

**IMPORTANT:**

* If you want to test your program, you can implement a **main()** function given in the stub and you can use **RUN CODE** to test your main() provided you have made valid function calls with valid data required.

##### Execution time limit

10 seconds

REPORT AN ISSUE

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

class Activity{

//Implement Activity class here..

}

public class Source {

//Implement the two function given in description in here...

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

//Write your own main to check the program...

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*