# Advanced Programming Tute/Lab - Exceptions and Files

### Objective: The objective of this tutorial is to refresh exceptions, files and inner classes.

**Review Questions:**

1. What are benefits of inner classes?
2. How does private inner classes make the OO deign more maintainable?
3. What the benefits of anonymous classes?
4. What is an Exception and what does it represent?
5. How do you handle an exception in a program where it occurs?
6. What is exception propagation and how does it work?
7. How do you create and use custom designed exception types? What are the benefits?
8. How do you open text files for reading and writing?
9. How do you read different types of data from a text file (ie. strings, numbers, etc)?
10. How does random access file differ from text files?
11. What do you do with text files after you have finished reading from or writing to them?

# Tutorial Exercises (Exceptions):

1. a) What is the output of the following code?

int theValue = 30; try

{

System.out.println("Block entered"); if (theValue > 10)

throw new Exception ("Value exceeded"); System.out.println("Leaving try block");

}

catch (Exception e)

{

System.out.println("Exception occurred: "

+ e.getMessage());

}

System.out.println("After catch block");

b) What is the output if we change the top line to int theValue = 0;

1. Consider the following code:

try

{

statement1; statement2; statement3;

}

catch (Exception1 e1)

{

………

}

catch (Exception2 e2)

{

…………

};

finally { statement4;

}

statement5;

### If statement2 causes an exception, decide whether the following are true or false:

* 1. statement3 is executed
  2. if the exception isn't caught, statement5 is executed
  3. if the exception is caught in the catch clause, statement5 is executed
  4. statement4 is executed

1. Assume SubException extends Exception. Consider the following code:

try

{

statement1; statement2;

}

catch (Exception e)

{

………

}

catch (SubException se)

{

…………

};

### What happens if any of the statements throws a SubException? Is there something wrong with this code?

**Tutorial Exercises (Files):**

Complete the program below to read any file and print all the lines containing the specified word. For example, if you give a .java file and the word as “public” then the program should print only the lines containing public.

Hint: The readLine() method of BufferedReader will return null when end of file is reached. The String indexOf(String match) method can be used to find if a string contains a word.

import java.util.\*; import java.io.\*; public class TestFile1

{

public static void main(String args[]) throws IOException

{

Scanner console = new Scanner(System.in); System.out.print("Enter file name : "); String fName = console.nextLine(); System.out.print("Enter string to match : "); String match = console.nextLine(); BufferedReader br =

new BufferedReader(new FileReader(fName)); String s;

System.out.println("Matching lines are "); while ( … )

{ ...

}

}

}

1. You are given the file marks.txt, containing the student marks for a number of tests in the format below. Each student in a class may have 0 or more test results and the number of students in the class may vary. If the student has not taken any test the corresponding line will be blank in the file as shown below. Write a program to read the file, compute the average marks and print them in the format below. Note, the student name must be shown even if he or she has not taken any tests.

Marks.txt (input file)

### Expected Output

No marks

Abraham Lincoln 10 6

Charles Dickens 8 9 10

Paul Chen

David Norris 4 6

Peter McDonald 8 10 8 10

Enter file name : Marks.txt Abraham Lincoln : 8.0 Charles Dickens : 9.0 Paul Chen :

David Norris : 5.0 Peter McDonald : 9.

