**C S CI 2220 Introduction to data structures**

**Lab 1: Eclipse tutorial, debugging & file operations**

# Learning objectives

Objective 5 and Objective 7 from the course syllabus.

* Objective 5: Use basic Java program constructs and skills which include reference variables, basic data type variables, exceptions, file operations, generic programming.

If you did not take any prerequisite courses, you may want to check the Java overview file posted in week 1 module on Canvas to get an understanding of Java.

* Objective 7: You will need to learn how to use a new IDE, Eclipse and debug programs.

# Requirements

## Tasks

* Get familiar with Eclipse IDE.
* Use the IDE to debug programs
* Get familiar with Java file operations.

## Detailed instructions for program design and implementation

1. (20 points) Get familiar with Eclipse IDE.
2. Download and extract Lab1.zip file from week 1&2 module on Canvas.

Follow the steps in the Eclipse tutorial located in Lab 1 folder that you downloaded from canvas

and load Welcome.java and addition.Java from Lab1 into the (default package) in Eclipse. Run the two files.

Create a text file named eclipse\_test.txt to keep running results.

* 1. (10 points) For welcome.java, after Line 8, add one line to print the current system time. The function for getting the current system time is System.currentTimeMillis() Copy the running results to eclipse\_test.txt.
  2. (10 points) For addition.java,
     + Change the variable number to 20, copy the running results to eclipse\_test.txt.
     + Keep the variable number as 10 and change Line 14 step++; to

step \*=2;

Copy the running results to eclipse\_test.txt.

1. File operations.

Open core\_dataset.csv(a data file that was obtained from <https://www.kaggle.com/rhuebner/human-resources-data-set>)[.](https://www.kaggle.com/rhuebner/human-resources-data-set) It is a comma-separated values (CSV) file. The first row of the dataset is the metadata, showing what information of an employee is recorded. Each row contains information about one employee. Have a quick look at the content.

Create a java file named EmployeeFileOp.java under the default package. In this java file,

* 1. (30 points) the code should be able to

1. open core\_dataset.csv for reading,
2. open a new file old\_employee.csv for writing,
3. read each row of the employee information from the given input file,
4. extract the Employee Number and Age information of each employee, and
5. Write the Employee Number and Age information of an employee with age **older than 50** to a file named old\_employee.csv. Note that the old\_employee.csv does not need to keep the first row of the dataset. Please properly design your program with proper methods.  
   1. (10 points) Create a main function to properly call the read and write functions that you created.
6. Code debugging.

(40 points) You are going to analyze a program that displays Pascal's triangle. This triangle is a sequence of integers that arises in numerous areas of math and computer science, especially combinatorics and probability. For example, the Pascal triangle of height 4 is:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Entries in Pascal's triangle are indexed by integers. n is the number of the row and k is the position from the leftmost member of the row. The indexes in both directions start at zero (0), so in the last row listed above, C(4,0) = 1, C(4,1) = 4 and so on.

The values themselves are computed by the formula C(n, k) = , which is called a combination. n! denotes a factorial, that is, the product n\*(n-1)\*(n-2)\*...\*2\*1. The combinations can be interpreted as the number of ways to choose k elements from a collection containing n elements. When described, it is customary to say "n choose k", for instance '4 choose 2 is 6'.

Let’s look at this closer: If four objects are numbered 1 through 4, think of how many ways can you combine two of them. This is what "n choose k" calculated for us.

(5 points) List all the possible pairings here. (Ex. {1, 2}, {2, 3}, …) Keep in mind that the order of the numbers in a pairing does not matter – {1,2} is the same pairing as {2,1}. After you list all the pairings, compute C(4, 2) by using the above formula. What is the result? Does it match the number of pairings you listed? (Hint: answer to the last question should be yes)

* + - 1. (1,2)
      2. (2,3)
      3. (3,4)
      4. (1, 3)
      5. (1, 4)
      6. (2, 4)

C(4,2) = 4!/ 2!(4 – 2)!

C(4,2) = 4 \* 3 \* 2! / 2!(2)!

C(4,2) = 4\* 3 / 2

C(4,2) = 2 \* 3

C(4,2) = 6

## Debug the Program Code

(35 points)Load PascalTriangle.java and PascalTriangleTester.java in the default package(Files are located in Lab 1 on Canvas) into the default package in Eclipse. Compile and run them.

1. What output do you get when you request a triangle height of 5?

Enter height: 5

1

4 2 1

576 144 24 4 1

Enter height: 5

1

4 2 1

576 144 24 4 1

1. How many rows should have been generated for a height of 5?

6 rows should have generated (0 1 2 3 4 5)

By now, it’s obvious that there is a problem. Let’s start investigating by setting a breakpoint at the line:

skip(spacesToSkip); // space to make a triangle

in the PascalTriangle constructor.

1. Debug the program (height is 5). What is the value of n when the breakpoint is reached?

The value of n when typing is n = 0

1. Run the program until it reaches the breakpoint again. What value do you expect n to be and what is the debugger reporting?

Expected: I expecet the value of n = 0 or n = 1

Actual: its skipping to n = 2

1. Once again, run the program until it reaches the breakpoint again. What value do you expect n to be and what is the debugger reporting?

Expected: expected shoud be n = 1 or n = 2

Actual: The actual is n = 4

1. The variable n is supposed to take the values 0, 1, 2, 3, 4, 5. Find the problem and fix it. What did you do to fix it?

I found the problem there was a n++; at the end adding an extra one each time making it count from

2 4 6 8 etc. breaking the code. I removed it and now it counds up by 1 normally.

1. Run your corrected version again with a height of 5. You should now have six rows of output, but the values are still wrong. What values do you get? How do you know they are wrong?

Enter height: 5

1

1 1

4 2 1

36 12 3 1

576 144 24 4 1

44002880 360 40 5 1

I know they are wrong because the outside numbers aren’t 1

To determine why the values are wrong, set a breakpoint at the line

return comb;

in the combination method (you can remove your prior breakpoint). Debug your program until the method is executed with the values n = 3 and k = 1.

1. What should the value of comb be? (Calculate C(3,1) from part 1 of this lab.) What is the actual value?

Expected: C(3,1) = 3! / 1!(3-1)!

= 3 \* 2 \* 1! / 1! (3 – 1)!

= 3 \* 2 / 2

= 3

Actual: I am getting Comb = 12

1. Check why the value is computed incorrectly, and fix the computation. What did you do?

The value was incorrect because the entire bottom has to be ( ) so once I added the ( ) to the denominator

It fixed it and foxed

1. After fixing the error, run the test again (without breakpoints). What is the output?

Enter height: 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

# Submission instructions

• Submit through canvas a zip file consisting of eclipse\_test.txt, EmployeeFileOp.java, old\_employee.csv, PascalTriangle.java and PascalTriangleTester.java. Please do NOT submit the .class files.

# Grading criteria

1. The score allocation is already put in the questions.
2. Please make sure that you test your code thoroughly by considering all possible test cases.
3. 5 points will be deducted if submitted files (including file types, file names, etc.) do not follow the instructions.
4. At least 20 points will be deducted if your code cannot be run on CS servers.

2