

---

## CS 251: Code warrior: Java and multi-core programming (Inlab)

- Due: 8:10 PM 04/10
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

### Overview

Java is the world's #1 programming language, so says Oracle :). Java is characterized by the buzzwords: Simple, Object oriented, Distributed, Multithreaded, Dynamic, Architecture neutral, Portable, High performance, Robust, and Secure. The official language for Android development is Java. For more information on learning Java, visit [java oracle page](#) The goal of this lab is to learn tiny, yet powerful aspects of Java along with a basic overview of an IDE. We won't be learning the entire language :)

### A0. Bignum in JAVA

Numbers are represented in C++ using int, long (or float, double). But these classes have a limited range. In some cases, we might need to handle numbers outside the range of these basic classes. In such cases, we will have to write our own classes in C++ or use external classes such as GMP. Java has built in arbitrary precision. Bignum (BigInteger and BigDecimal). Bignum has no limits and can handle any number.

0. Harry is trying to stop Snape (or so he thinks) from stealing the philosopher's stone. Harry has crossed the potions task. Next he is faced by a math task. Help him solve it so that he can stop Snape from stealing the stone. The task is to take numbers from a parchment and calculate the sum of combinations (  ${}^m C_n$  ) as described below
1. Write a Java program `math.java`. The program must read a file `input.txt` present in the same folder. The `input.txt` file's first line gives the number of cases (t). Each of the next t lines provides two integers ( $m_i, n_i$ ). Your program should print t lines of output. Each line must be the following sum:

$$\sum_{j=n_i-5}^{n_i+5} {}^{m_i} C_j$$

2. Redirect the output to a file from the program itself. Name the file as `output.txt`
3. Errors have to be handled elegantly using Java's built-in mechanism. The errors can be negative integer inputs, decimal (non integer) inputs or  $m < n+5$  or  $n < 5$  in the input. Handle all errors mentioned above. You need to print a line explaining the error instead of the sum. Use descriptive information with no spelling errors :)
4. You can find a sample `input.txt` for which ensure that your code produces right results

### A1. Eclipseing the Opposition

**Why Eclipse?** Eclipse is a platform that has been designed from the ground up for building integrated web and application development tooling causing rapid development of integrated features based on a plug-in model.

0. Back to the plot. Harry is happy that he solved the task but the actual task set up by Dumbledore is always out of the box. <sup>1</sup> His hologram (well, magic right!!) said to Harry that now that you have written `math.java` in your favorite editor, its time to switch to Eclipse. You now have to use the Eclipse IDE to execute `math.java`. Write the steps you carried out for doing the same (how will you import ?) in a text file : `A1.txt`. Where does eclipse place the source files? The executables? What is the classpath? Submit `A1.txt`

## B. Threading the pieces together

Writing a program with threads is more involved (and we have the additional burden of making it thread-safe!). Still, threading has advantages which you will explore now.

Harry knows that the Chamber of Secrets password is the index of cell<sup>2</sup> which has Voldemort's key within it. Harry has been given `thought.java` which waits for a huge nested loop to complete before it prints the answer element in the count array. Harry's intuition told him that the priceless Voldemort key is 3523. (We don't know Harry's thought process but hey, that was his intuition.)

0. Modify your code in such a way that each outer loop iteration spawns a new thread after generating the required random number. The thread takes in all the required variables along with the random number. Each thread does the specified task of the inner loop and assignment for the outerloop array (cell) along with checking whether the cell has the Voldemort key. When you find the Voldemort key, you should print the cell index to which the key is assigned. (Important note says J.K. Rowling: The cell password is useless unless the work of the inner loop is done.)
1. Start with analysing `thought.java`. Find the time taken and CPU usage (in terms of percentage) for the plain vanilla version of `thought.java` and provide us with the screenshots. Read about `top` and `time` (real time) commands of bash.

Harry now understood that this straight forward thought process is not good.

2. Using eclipse, create another java file : `t.java`: Copy the contents of `thought.java` into `t.java`. We want to move the working code (the inner loop) into the task of each thread. Eclipse provides a feature to replace a piece of code into a function call using a single click (ok maybe two). How? Use this feature to replace the inner `for` loop in `t.java` file with a function call `run()`. Write the feature you used in `B.txt`

Place this new function in a separate class, and make sure that the necessary plumbing is present between the main method and the new class. Use the name `xyz` for this class. Your program should now work and give the answer, so give yourself some time to debug this step before proceeding further.

3. Generally the definitions of the classes invoked in the main class should be defined in a separate java file (why?). Hence now create another java file and define the `xyz` class used in `t.java` in this new file (and of course deleting the definition from `t.java`)
4. `xyz` is not a nice name. Eclipse provides with another feature that can change names of variables, functions or classes to any name that you want across **all** files inside a project. (how?). You now need to change the name of the `xyz` class that you have called inside the main method in `t.java` (and defined in the new file you created) to `Loop` using this feature. Write what feature you used for this and how did you use it in `B.txt`
5. Submit the modified code as `thought_thread.java` along with the other java files required. Make sure the whole system works outside eclipse. Also provide an explanation on how your modified program is superior to the originally given program. Be specific, for example, give

---

<sup>1</sup>Dumbledore was one of the persons who placed tasks on the way to the Philosopher's stone.

<sup>2</sup>cell indices start with zero

timing information, in addition to any insights you got in this assignment. Provide us with the screenshots of `top` and `time` commands in the submission for the run of `thought_thread.java`. Push all this information including (tiny) screenshots if any in your submission. You should submit all relevant java files only but not any class files.

## C. Multiple Threads and Multiple cores

Threading and multi-core programming are two techniques used for the efficient usage of CPU for the programs we write. Java has threads as a built-in feature, but that's no good for exploiting multiple cores. Using the invisible cloak as a desperate measure Harry went to the restricted section of the library and read about multi-core programming. We will use an open source **OpenMP**-like preprocessor called **omp4j** for multi-core programming.

0. Download the v1.2 jar from the website and setup a permanent alias as mentioned in the site. Write the command/steps used to setup the permanent alias in the text file named `C.txt`.
1. Back to `thought.java`. Use `omp4j` and use the OpenMP directives (be careful in choosing the directives, we need the same answer as previous cases) to parallelise the outer for-loop. Name the file as `thought_multi_core.java`. And write the time and CPU usage obtained for the code in `C.txt` and provide us with the screenshots for the same in the submission
2. Add your inferences about the observations made in `C.txt` and give the final verdict on the thought mechanism for Harry for the given problem

Harry will need your help in outlab too, as we have 5 more books leftover to finish. Thank you muggles for the help :)

## Submission Guidelines

1. When you submit, please document individual percentages such as Student 1: 80%, Student 2:100%, Student 3:10% in the `readme.txt`. In this example, the second student will get full marks (10/10) and the first student will receive 8/10.
2. Do include a `readme.txt` (telling us whatever you want to tell me). (The reflection essay is not required for inlab, but is required for outlab.) Do include group members (name, roll number), group number, honour code, citations etc.
3. The folder and its compressed version should both be named `lab09_groupXY_inlab` for example folder should be named `lab09_group09_inlab` and the related `tar.gz` should be named `lab09_group09_inlab.tar.gz`.
4. The submission folder should contain 4 subfolders `taskA0`, `taskA1`, `taskB`, `taskC`
5. The subfolder `taskA0` should contains 1 file `math.java`
6. The subfolder `taskA1` should contain `A1.txt`
7. The subfolder `taskB` should contains file `thought_thread.java`, screenshots, `B.txt` and other relevant java file(s).
8. The subfolder `taskC` should contain 3 files `thought_multi_core.java`, screenshots and `C.txt`
9. Your submission folder should look something like this:

```
lab09_groupXY_inlab
├── readme.txt
├── taskA0
│   └── math.java
├── taskA1
│   └── A1.txt
├── taskB
│   ├── thought_thread.java
│   ├── Other java files
│   ├── Screenshots
│   └── B.txt
├── taskC
│   ├── thought_multi_core.java
│   ├── Screenshots
│   └── C.txt
```

## How We will Grade You [30 Marks]

### 1. Task A0 [10 Marks]

- `math.java` with the sample `input.txt` giving required answers : 6 Marks
- Hidden Input (your code will be evaluated against this) : 4 Marks

### 2. Task A1 [3 Marks]

- `A1.txt` : 3 Marks

### 3. Task B [10 Marks]

- `thought_thread.java` and other java files : 5 Marks
- Screenshots : 3 Marks
- `B.txt` : 2 Marks

### 4. Task C [7 Marks]

- `thought_multi_core.java` : 4 Marks
- Screenshots : 1 Mark
- `C.txt` : 2 Marks