<div align="center">

CSIT121/821 Lab Exercises

Lab 1

Deadline: 23 August 2020, 10 PM

</div>

Objectives
- Familiar with java
- Review fundamental object-oriented programming

An exercise (The exercise will not be marked. You can jump to the Tasks if you already familiar with Java JDK)

Step1: Installation of Java JDK, and Text Editor.

1. Please use the following link to download and install Java JDK on your laptop. Please make sure you download the correct installation file for your operating system:
https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

2. Please use the following link to download the Text Editor.

      - NotePad++ for Windows user: https://notepad-plus-plus.org/downloads/
      - TextMate for OS user: https://macromates.com/download

Step 2: Setup the path to run the JDK tools.

The point of setting the environment variable is to let programs know in which directory executables like javac can be found.

1. Open Advanced System Settings

In Windows 10 press Windows key + Pause Key, This will open the System Settings window. Go to Change settings and select the Advanced tab. Alternatively, open "Windows search" – you will find it next to the Windows logo
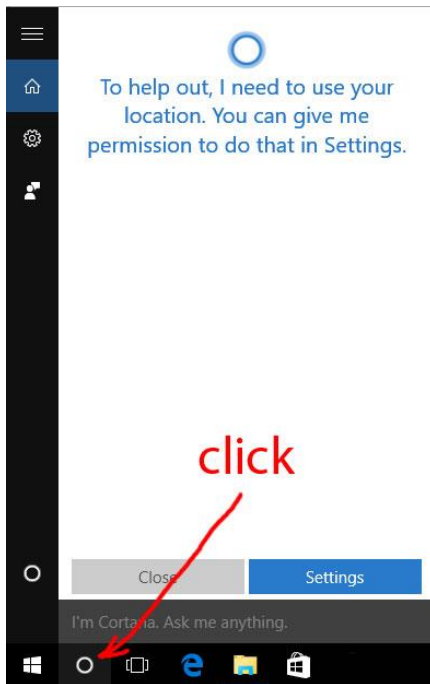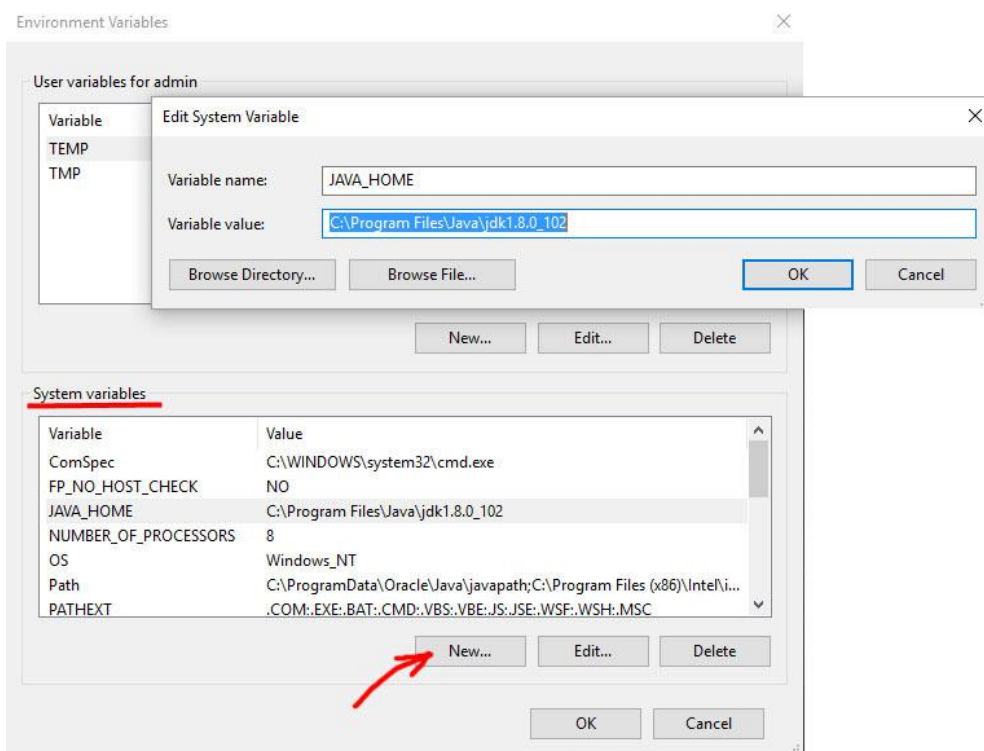
1. In the search field type in – advanced system settings
2. Click on the match on top of the list
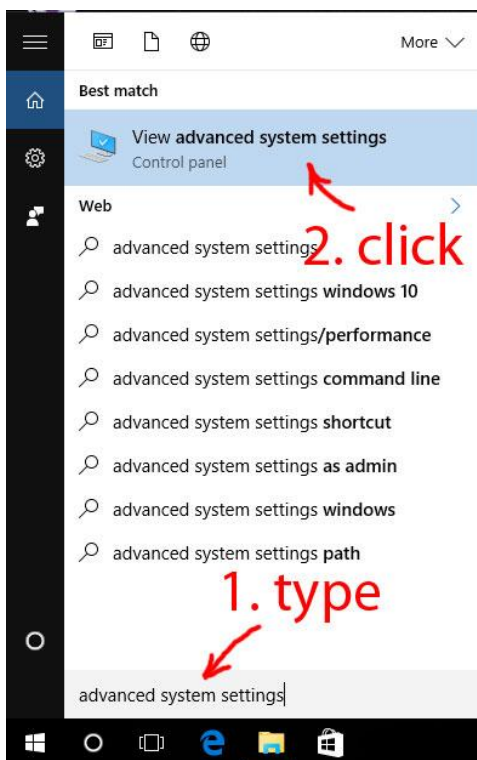
2. Set JAVA_HOME Environment variable
In "System Properties window" click "Environment Variables…"
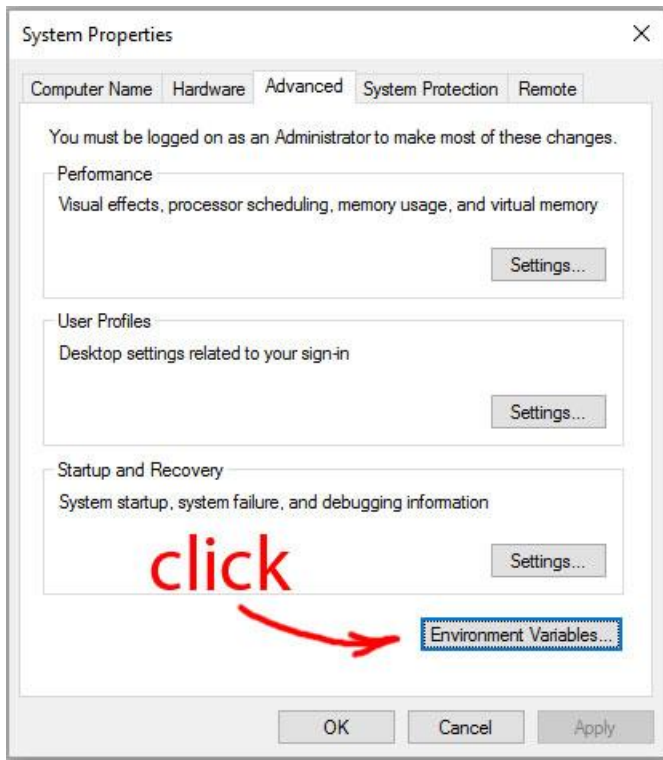
Under "System variables" click the "New…" button and enter JAVA_HOME as "Variable name" and the path to your Java JDK directory under "Variable value"

3. Update System PATH
1. In "Environment Variables" window under "System variables" select Path
2. Click on "Edit…"
3. In "Edit environment variable" window click "New"
4. Type in %JAVA_HOME%\bin
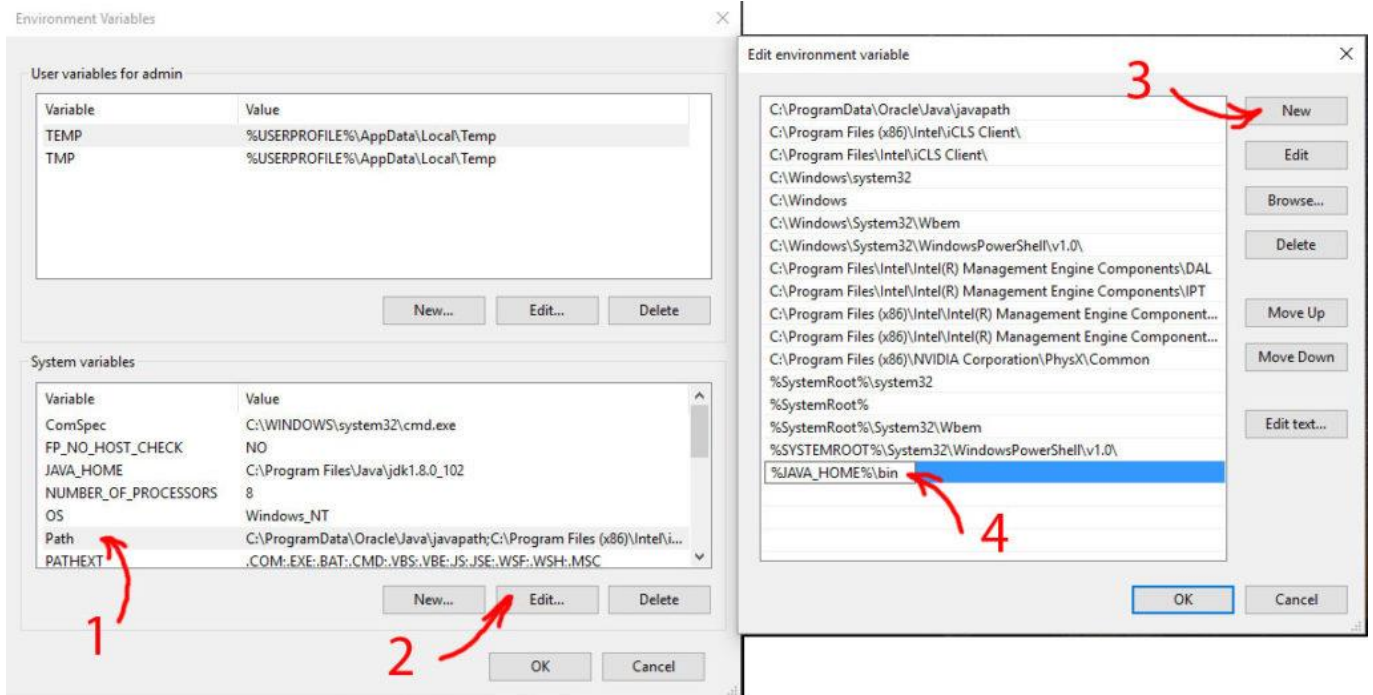
4. Test your configuration
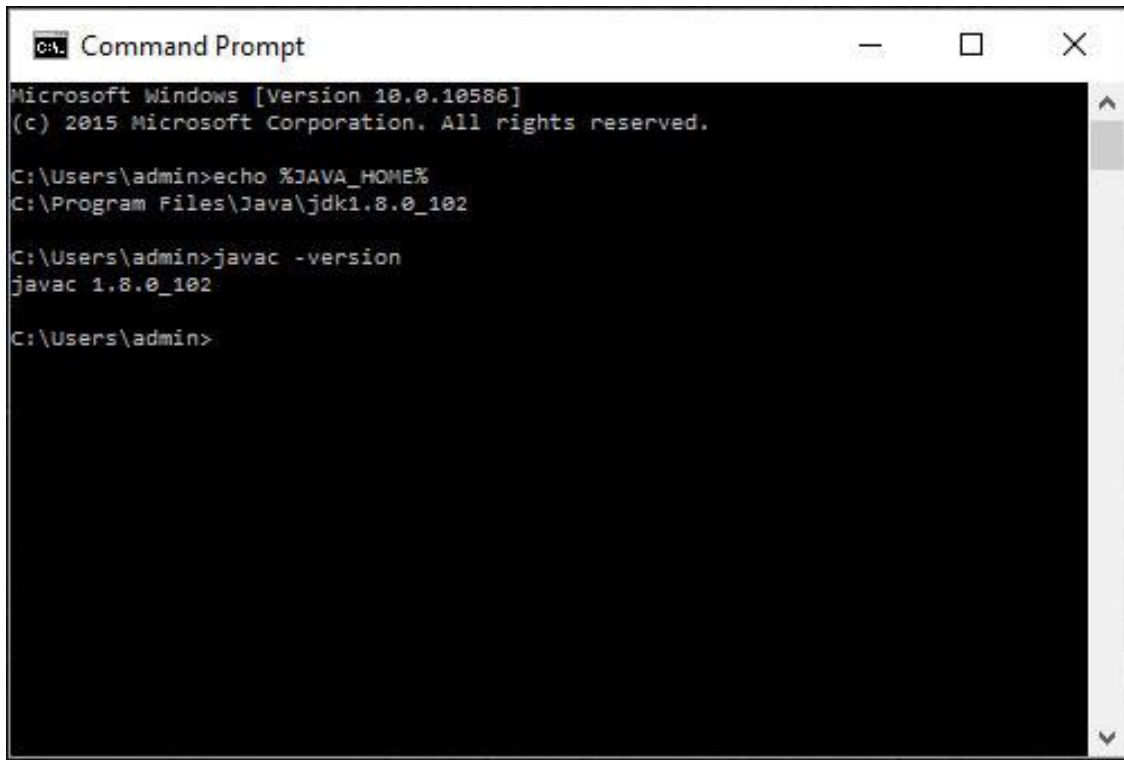Open a new command prompt and type in:

1. **echo %JAVA_HOME%**

this will print out the directory JAVA_HOME points to or empty line if the environment variable is not set correctly

Now type in:

1. **javac -version**

this will print out the version of the java compiler if the Path variable is set correctly or *"javac is not recognized as an internal or external command…"* otherwise

For Mac, please see the details here: https://www.java67.com/2015/11/how-to-set-javahome-path-in-mac-os-x.html

Step 3: Try your first Java program, HelloWorldApp.java

1. Open you text editor, create a new file and save it as HelloWorldApp.java;

2. Please type the following codes character by character. Don't copy-paste from this page. Remember that errors are your friend at this stage. Try to encounter common errors as early as possible to learn how to fix them.

```
class HelloWorldApp{
  public static void main(String[] args)  {
        // Display "Hello world!"
        System.out.println("Hello world!");
  }
}
```

3. Open the command line and goes to the folder where you save your java source code;

4. Compile the HelloWorldApp.java file by typing "javac HelloWorldApp.java". If your code has no error, the executable file "HelloWorldApp.class" will be added in the same folder; If there are mistakes in your code, please follow the hints given by the complier and correct the mistakes (debug) in your text editor and repeat this step;

5. Run your program by typing "java HelloWorldApp", then you will see "Hello World" print on your screen. You can modify the display content of your program once your program can be executed. For each modification, you must save the changes and restart from Step 4.

Tasks (All tasks will be marked. Five marks in total.)

A course management system contains three class, i.e., the Student class, the Subject class, and the Course class (the primary class).

The Student class is used to keep and maintain the personal information of a student. Its field data shall contain some basic information about a student, such as the student name, DOB, sex, student number and etc. With the Student class, Student objects can be created. The Student class shall also provide some mutator and accessor methods to modify and access the field data of Student objects.

The Subject class is used to keep and maintain a subject's information. Its field data shall contain the subject name, code, session and year, and a student list. The student list shall contain all students enrolled into the subject, and it can be implemented via an ArrayList of Student objects. The Subject class shall also provide the mutator and accessor methods to modify the subject information. More important, it shall contain methods to modify the student list, i.e., to enrol and withdraw students to the subject.

The Course class is the primary class of the system and is used to manage all subjects and students of a particular course. Its field data shall contain the course name, a student list (i.e., all students enrolled into this course) and a subject list (i.e., all subjects that the course contains). The Course class shall provide some methods to access and modify the course name, and the student and the subject lists.

Task 1: Class design with UML class diagrams (2 marks)

Based on the above description, you shall complete the design of the course management system with the three classes and draw UML class diagrams (with the UMLet or other professional tools, no hand drawing) to represent your design. You shall use the correct notations in the UML class diagrams and clearly show the field data, methods and associations between the three classes.

Task 2: Implementation (2 marks)

Implement your course management system by using Java. The implementation of all classes shall consistent with your UML diagrams. For the testing purpose, please using the following information and procedure in the *main()* method.

Course Name: Bachelor of Computer Science

Students (4 students): Amy Bell (Female, 01/01/2001, 100001), Bob Brown (Male, 02/02/2002, 200001), Cindy Ma (Female, 03/03/2001, 100003), and David Hintz (Male, 04/04/2000, 100004)

Subjects (2 subjects): CSIT111 (Programming Fundamentals, Spring 2020), CSIT121 (Object Oriented Design and Programming, Spring 2020)

1. To create the Bachelor of Computer Science Course object, four Student objects, and two Subject objects;
2. To add the four Student objects into the student list of the BCS Course object, and the two Subject objects into the subject list of the BCS Course object;
3. To enrol Amy, Bob and Cindy into CSIT111 and David into CSIT121, and display the course enrolment information as follows:

Course Name: Bachelor of Computer Science

Subject Name: Programming Fundamentals (CSIT111, Spring 2020)
Enrolled Students:
Amy Bell   (100001)
Bob Brown   (100002)
Cindy Ma   (100003)

Subject Name: Object Oriented Design and Programming (CSIT121, Spring 2020)
Enrolled Students:
David Hintz   (100004)
----

4. To withdraw Cindy from CSIT111 and re-enrol her into CSIT121 instead, and display the course enrolment information as follows:

Course Name: Bachelor of Computer Science

Subject Name: Programming Fundamentals (CSIT111, Spring 2020)
Enrolled Students:
Amy Bell   (100001)
Bob Brown   (100002)

Subject Name: Object Oriented Design and Programming (CSIT121, Spring 2020)
Enrolled Students:
David Hintz   (100004)
Cindy Ma   (100003)

Task 3: Compilation and Testing (1 marks)

Compile and test your program and make sure the outputs are same as the requirements.

Submission:

- Please submit your solution to Moodle. Email submission is not accepted.
- Please submit an **individual PDF document** to contains all your solutions for the three tasks.
- For Task 1, please include your UML class diagrams.
- For Task 2, include your Java codes.
- For Task 3, please include the snapshots which clearly shows the compilation and the execution of your program in each step listed in Task 2.

Note: Plagiarism will be reported to the school, and all involving students will receive the zero mark.