Introduction

Hello folks! Welcome to CSE 8B this quarter! This is your first assignment so it will not be too hard. The main purpose of this assignment is for you to be aware of the importance of Academic Integrity, get familiar with basic command-line navigation and operations, prepare your development environment for this quarter, and compile and run your first Java program in CSE 8B. This assignment is partially adopted from open source materials.

Note: You should complete this assignment individually.

Part 1: CSE Academic Integrity Tutorial

UCSD CSE department is not tolerating any violation of Academic Integrity. Unlike courses from other departments, you are required to submit code, and there are specific programming qualifications you need to know before you write any code. Most of the time, students violate academic integrity just because they don’t know it was a violation. The below tutorial is your one-stop shop of academic integrity to avoid such situations.

1. Please watch the Source Code Plagiarism tutorial thoroughly.
2. After you finish watching the tutorial, you must complete the CSE Integrity of Scholarship Agreement. By signing this, you acknowledge that you have completed and understood the tutorial and also the agreement.

Part 2: Command-Line Navigation

Now let us introduce you to your new friend in programming: the command line interface! You might have seen programmers using a black window and typing commands on it. That is the command line interface and every programmer is using that. Throughout this quarter, you will be using the command line interface to navigate, compile and run your program for all assignments. It has many names, such as cmd, CLI, console, or terminal.

Open the CLI

First, let’s open the CLI.

- Mac OS:
  - Go to Applications → Utilities → Terminal.
Search “Terminal” in the Spotlight Search bar.

Linux: We’re using UCSD Linux Cloud as the example. You can find it under Applications → System Tools → Terminal. (Picture next page for your reference)
- **Windows**: Depending on your version of Windows and your keyboard, one of the following should open a command window (you may have to experiment a bit, but you don't have to try all of these suggestions):
  - Go to the Start menu or screen, and enter "Command Prompt" in the search field.
  - Go to Start menu → Windows System → Command Prompt.
  - Go to Start menu → All Programs → Accessories → Command Prompt.
  - Go to the Start screen, hover your mouse in the lower-left corner of the screen, and click the down arrow that appears (on a touch screen, instead flick up from the bottom of the screen). The Apps page should open. Click on Command Prompt in the Windows System section.
  - Hold the special Windows key on your keyboard and press the "X" key. Choose "Command Prompt" from the pop-up menu.
  - Hold the Windows key and press the "R" key to get a "Run" window. Type "cmd" in the box, and click the OK key.

![Command Prompt](image)

**Prompt**

Now you should see a black or white window that is waiting for your command.

- **Mac OS or Linux**: you probably will see a `$` or `%` and a space on your current line.
- **Windows**: you probably will see a `>` on your current line.

Each command will be prepended by one of the characters above, but you should not type it by yourself. The terminal will do that for you. Also, there might be something like `C:\Users\name` or `name@MacBook` before the above characters. This is totally fine.

**Your first command**

Now let's type the following command, no matter which OS you are using. Keep in mind that you should not type any leading characters by yourself.

- **Mac OS or Linux:**
$ whoami

- Windows:

  > whoami

  Then hit enter. And you can see your username appear.

- Mac OS or Linux:

  $ whoami
  myusername

- Windows:

  > whoami
  computername\myusername

**Current directory**

If you want to know where you are right now, or the full path of your current directory, you can try the following commands.

- Mac OS or Linux:

  $ pwd
  /home/linux/ieng6/cs8bfa20/myusername

  "pwd" stands for "print working directory". The above result is from UCSD Linux Cloud.

- Windows:

  > cd
  C:\Users\myusername

  "cd" stands for "change directory".

**List files and directory**

If you want to know what is in your current directory, you can try the following commands.

- Mac OS or Linux:

  $ ls
  Desktop   Downloads   Pictures   Templates   ...

  Desktop   Downloads   Pictures   Templates   ...

  "ls" stands for "list directory".
- **Windows:**

```plaintext
> dir
Directory of C:\Users\myusername
10/08/2020 07:28 PM <DIR>      Applications
10/08/2020 07:28 PM <DIR>      Desktop
10/08/2020 07:28 PM <DIR>      Downloads
10/08/2020 07:28 PM <DIR>      Music
...
```

**Change current directory**

Now let's go to our Desktop directory.

- **Mac OS or Linux:**

```plaintext
$ cd Desktop
```

- **Windows:**

```plaintext
> cd Desktop
```

If you want to check whether you have successfully changed the directory, use the “pwd” or “cd” command we mentioned above to check your current directory after changing.

**Create directory**

It would be nice if you can create a CSE 8B directory on your desktop for file management.

- **Mac OS or Linux:**

```plaintext
$ mkdir cse8b
```

- **Windows:**

```plaintext
> mkdir cse8b
```

If you want to check whether you have successfully created the directory, use the “ls” or “dir” command we mentioned above to check your current directory’s content.

**Practice time!**

Based on the above tutorial, please create a new directory named `will_be_deleted` inside of the `cse8b`, then change your current directory to the `will_beDeleted` directory.
Create a Java file

Now that we’re inside the `will_be_deleted` directory, we want to create a java file named `deleteIt.java`. Here’s what we should do.

- Mac OS or Linux:

```
$ vim deleteIt.java
```

After this command, your Vim editor is on. But you are not able to type because you’re in normal mode, the default Vim’s mode. In order to make any input, you need to press the key `i` first. Then there will be a `-- Insert --` at the bottom left of the window. If you want to quit the Insert mode, you need to press the key `esc`. After that, if you want to save the file, you can type `:w` directly. Then if you want to exit/quit the file and go back to your command line, you can type `:q` directly. The above two operations can be combined to `:wq`. If you’re using UCSD Linus Cloud, you can also give `gvim` a try. Just substitute vim by gvim in the command prompt.

- Windows:

```
> notepad deleteIt.java
```

The notepad is more straightforward. You can just type whatever you want, save the file, and close the file if you want to exit/quit.

We’re using Vim for Mac OS/Linux and notepad for Windows as examples. You can definitely use your favorite text editors or IDEs, but we’re not covering other options here.

Delete a file/directory

Before we delete anything, let's try to go back to `cse8b` directory.

- Mac OS or Linux:

```
$ cd ..
```

- Windows:

```
> cd..
```

Using `..` with the `cd` command will change your current directory to the parent directory. That is, the directory that contains your current directory. You can check where you are now by one command we mentioned above.

Now we can delete the directory and its contents all at once.

- Mac OS or Linux:
$ rm -r will_be_deleted

The `-r` represents recursive. The “remove recursively” operation is especially for removing directories. It will remove everything inside of the `will_be_deleted` directory.

- Windows:

  > rmdir /S will_be_deleted

You can check whether the directory is removed by one command we mentioned above. However, all remove commands like `del`, `rm`, `rmdir` are irreversible. Think twice before you do any deleting.

**How to get help?**

One last thing before you go to the next part, we want you to know how to call out the manual for each command. Let’s say we want to learn more about the `cd` command

- Mac OS or Linux:

  $ man cd

- Windows:

  > cd /?

Then you can see a detailed manual for this command you put. If you want to explore more, you can take a look at this website: [http://man.he.net/](http://man.he.net/).

**Part 3: Compile and run some code!**

Last part of this assignment is for you to compile and run your first piece of code in CSE 8B. If you’re using UCSD Linux Cloud, Oracle JDK 8u261 is already installed and configured on your UCSD Linux Cloud class account so you don’t need to worry about the installation. However, if you’re working on your own computer, here’s the step to install.

First, please go to this Oracle link to download. You might need to register a Oracle account to do so.

- Mac OS or Linux:

  ![macOS x64](https://example.com/jdk-8u261-macosx-x64.dmg)

  Download the .dmg file. Run it and follow the instructions till the end.

- Windows:
There are two options for Windows. If you have a 32-bit machine, download the Windows x86 version. If you have a 64-bit machine, download the Windows c64 version. After finishing the download, run the downloaded .exe file. You will see the following window. **Please take a note of the full path you are installing to.**

![Java SE Development Kit 8 Update 251 - Custom Setup](image)

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel.

- Development Tools
- Source Code
- Public JRE

**Feature Description**
Source code for classes that comprise the public API of Java. The source code requires 27MB on your hard drive.

Install to:
C:\Program Files\Java\jdk1.8.0_261\  

Finish the current installation. Then please right click on the “This PC”.

![This PC](image)

Then click on the “Properties” of the pop up menu. Go to “Advanced system settings” on the left.
Then click on the “Environment Variables” on the bottom right.

In the “System variables” section, click on “path”, then click on “Edit”.

---

System Properties

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign in

Startup and Recovery
System startup, system failure, and debugging information

Environment Variables...
In the new window, click on “New” on the top tight, then paste the path that your JDK installed to plus bin. For example, as the screenshot above, my JDK was installed to \texttt{C:\Program Files\Java\jdk1.8.0_261}, so I put \texttt{C:\Program Files\Java\jdk1.8.0_261\bin} here. Then click on ok.
After this, click on “New” in the User variables section.

In the new window, put the variable name and the above path we used, i.e. `C:\Program Files\Java\jdk1.8.0_261\bin`. Then click on ok.

Then you click ok on the Environment Variables window.

Now you should have installed the right version of Java8 for this course. But how to check this? Please open your terminal and go to the cse8b directory first. Then run the following command.

- Mac OS or Linux:
  
  ```
  $ java -version
  ```

- Windows:
  
  ```
  > java -version
  ```

You should see the java version "1.8.0_261" now. Finally, please run the following command:

- Mac OS or Linux:
  
  ```
  $ java -version 2> java_version.txt
  ```

- Windows:
Now that we have Java8 installed, let’s try to compile and run a piece of code from command line.

First, please create a new Java file named `Assignment1.java` from the command line. You can put the java file in any directory. The `cse8b` directory we created is a good choice. Copy the following code snippet to that Java file.

```java
public class Assignment1 {
    public static void main(String[] args) {
        System.out.println("Hello CSE 8B!");
        //Put your code below
    }
}
```

Then you need to write a single line of code to print `My name is xxx` starting from a new line.

After that, save your code and go back to the terminal. Before you run every Java code, you need to compile it first.

- Mac OS or Linux:
  ```bash
  $ javac Assignment1.java
  ```
- Windows:
  ```cmd
  > javac Assignment1.java
  ```

Then you can run the program by the following command.

- Mac OS or Linux:
  ```bash
  $ java Assignment1
  ```
- Windows:
  ```cmd
  > java Assignment1
  ```

You should see two lines of output in the terminal. Last step, run the following command and **Turn in the pa1_output.txt to Gradescope.**

**You will find a new .txt file generated. Please check if that file contains the correct Java version information.**
Submission

Very important! Plz follow the instruction below carefully and make the exact submission format. This is important since we will use scripts to grade so if you don't follow the same submission format you probably will receive a zero.

1. Go to Gradescope and click on PA1.
2. Click the DRAG & DROP section and directly select the two required files (java_version.txt and pa1_output.txt). Drag & drop is fine. Please make sure you don't submit a zip. Just the two files solely. Make sure the names of the files are correct.
3. You can resubmit unlimited times before the due date. Your score will depend on your final submission, even if your former submissions have a higher score.
4. The autograder is for the use of the instructional team. You won't see the result of the autograder. As long as you uploaded your two files you're good to go.
5. Your submission should look like the below screenshot (order of file doesn't matter). If you have any questions, feel free to post!