Assignment 1

CSE205 Assignment1

Due Date: Thursday, January 16th, 5:30pm
Important: This is an individual assignment. Please do not collaborate. Do not share your solution with anyone.
Make sure to follow the academic integrity policies. It must be submitted on-line (course website)
20 points total (This assignment will be counted towards your grade).
No late assignment will be accepted

1. In this course, all programming assignments are done using Java programming language. You can use any operating systems (Windows, Unix,...), but we will be using version 8
Java compiler.
Please download the latest version 8 from the following website:
[https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html (Links to an external site.)](https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)
Downloading and installing this version might take some time (you might not see anything for one minute or more), so be patient and allow enough time to complete this installation. Choose to remove any older or newer version of Java compiler if the installation process asks you to do so. If you see a web browser popping out with "Verify Java Version", then the installation is
complete. If you do not install the version 8, then you might have some problem later in the semester in this course when we learn JavaFX. If you write a Java program using other Java compilers such as J++, it might not compile when it is submitted to the CSE205 course server.

Programming with Windows machines:
You can program using MS-DOS/NotePad, but some recommended user-friendly editor software for Java programing are TextPad, jGrasp, and Eclipse (Links to an external site.). TextPad provides a free trial version and jGrasp and Eclipse are completely free. All machines in the CSE205 labs are equipped with TextPad. Note that these software need to be installed after Java compiler is installed.

Programming with Mac:
Mac usually has a Java compiler in it already. You can use Eclipse (Links to an external site.) to edit your program, then compile/execute it. Or you can use any editor to write your program, then you can use Terminal (open 'Finder>Applications' or Command+A, then go to 'utilities' and you can find 'Terminal'.) to go to a folder where your program file is located, then compile and execute (please see the instructions below for this).

Programming with Unix/Linux machines:

If you would like to program in a Linux/Unix environment, the university provides an account in general.asu.edu. Subscribe to General UNIX Cluster Server. You need to visit the ASU Computer Accounts Self-Sub website: http://www.asu.edu/selfsub
You can login using your ASURITE & password. It will list what you are currently subscribed to and what other options are available for you to sign up.
Add
General
Computing Server if you don't have it.
After a few days you can verify that you have your General UNIX account ready. You can then log onto "general.asu.edu" server using your ASURITE user ID and your password.

2. Download Assignment1.java to your computer. The name of this file must be "Assignment1.java" with the upper case "A". If you submit a file with another name, it will cause an error when it is submitted.
This Java program reads an integer from a keyboard and prints it out with other comments. Modify the comments at the top of the program to reflect your personal information. Submit Assignment1.java on Gradescope->Assignemnt1 , which is provided on the course web site.

You will see that the program is not passing the test cases. Your program is tested with 4 test-cases (4 sets of input and output files). In order to pass all test cases, your program needs to produce the same exact output as the ones given below. (Even one character difference causes failure.) This is a part of Software Engineering Testing process (that we will be discussing in class). Download the following input files, input1.txt, input2.txt, input3.txt, and input4.txt, and the following output files, output1.txt, output2.txt, output3.txt, and output4, and save them in the same directory as Assignment1.java is located to test your Assignment1.java output files with the ones provided.
input1.txt
input2.txt
input3.txt
input4.txt

output1.txt
output2.txt
output3.txt
output4.txt
This program uses the Scanner class. The Scanner class simplifies the way programs read
input. If you are using Windows, choose Start->Run, then type "cmd". This opens up a MS-DOS. if you are using Mac, open 'Finder>Applications' or Command+A, then go to 'utilities' and you can find 'Terminal'. You need to use commands such as "cd" (change

directory) to change the directory to get to where Assignment1.java is located. If you are doing this in your home computer, you might need to configure your machine. Go to the directory where Assignment1.java is located and compile this Java program by typing:
javac Assignment1.java

In a DOS or a Terminal, you can use the command "cd" (change directory) to change your directory or folder:
cd foldername
will take you to the sub-folder in the current folder.
cd ..
will take you back to the previous folder (the folder that contains the current folder.)
Once you reach the folder where Assignment1.java is located, then you can type the above command (*javac ...*) to compile. Note that you can compile your program using any IDE such as TextPad or Eclipse. So if you can compile it with other means, then you don't need to use a DOS or Terminal in this step. Also, if you are using Windows, and "javac" command does not work, then you might need to configure your machine.
Then execute this program using the first test case by typing:
java Assignment1 < input1.txt > myout1.txt
Here the program reads from
input1.txt
file, and writes the output into the file named
myout1.txt

Note that you need to have Assignment1.java file and input1.txt file in the same folder.
Now we compare the output (myout1.txt) generated by this program and the expected output shown by the file output1.txt.

In Unix:
Type the following. It tells which lines are different in these two files (If two files are identical, there will be no message):
diff myout1.txt output1.txt
Alternatively, the following can be used as well:
cmp myout1.txt output1.txt
Based on this information, you need to modify the program ()Assignment1.java) so the program produces the same expected output. Repeat this process for other test-cases.

In Windows:
Download the file comparison tool FileCompare.jar (if this one does not work, you can find other file comparison software such as WinMerge (Links to an external site.) and UltraEdit (Links to an external site.) and compare two files, myout1.txt and output1.txt.
Modify the content of the program so the program produces the expected output. Repeat this process for other test cases. Resubmit the assignment and make sure to pass all 4 test cases. Other criteria such as documentation will be graded manually by TA/graders after the deadline of the assignment.

Grading Criteria:

5 pts - Documentation (header with your name, your information, and program description and comments within your code)

1 pt - Indentation and spacing (easy to read)

6 pts - Required classes/methods and functionalities implemented

8 pts - correct outputs/results

20 points total

What to turn in:

-Submit your modified Assignment1.java file using Gradescope-> Assignment1 provided on the course web site. Make sure that it is passing all test cases. Otherwise you will lose points for test cases (8pts).
