Assignment 3 - Team Maker & Word Count

Re-submit Assignment

Due May 29 by 11:59pm **Points** 50 **Submitting** a file upload **File Types** java

This assignment requires you to complete self-study of **Chapter 3**, especially the section of "**How to code simple control statements**" (p.78~83) about using while-loop and if-else statements.

Note 1: Boolean expression is a key part in writing both types of statements. Chapter 3 does not introduce more complex Boolean expressions, but, in this assignment, if you need to make a compound expression that consists of more than one Boolean expression, you will need to use Boolean operators "**&&**" for AND, and "**||**" for OR. To negate a Boolean expression, you add "**!**" to its front.

Note 2: If you need to completely stop looping at any point within a while-loop, you can write a 'break;' statement to exit. The execution continues on the statement after the while-loop statement, if there is one. See WhileExample2.java in May22.zip for how to use the 'break' statement.

Note 3: Reading and studying of Chapter 4 is not required by this assignment.

Submission

Different from Assignment 2, this assignment requires to submit only a single file called **Assignment3.java**, which can be done by either using Notepad++ and JDK in command line or using NetBeans. If you use NetBeans, you must copy this .java file to submit instead of the project.

Console (Sample output of the Assignment3 execution)

Fig-1

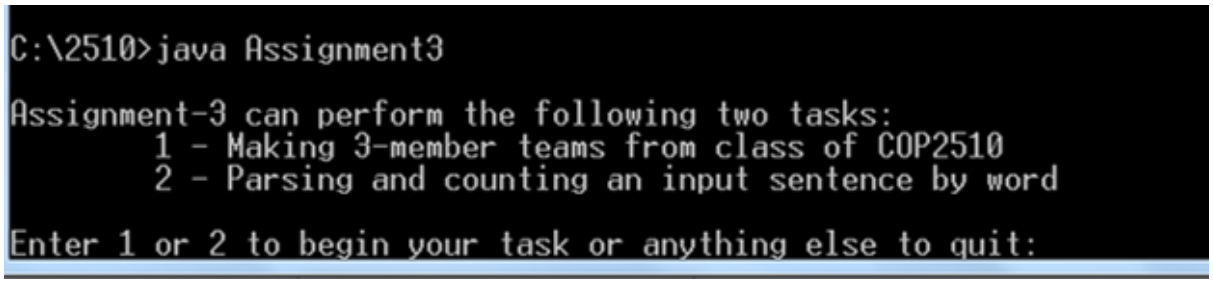

Fig-2

```
C:\2510>javac Assignment3.java

C:\2510>java Assignment3

Assignment-3 can perform the following two tasks:
        1 - Making 3-member teams from class of COP2510
        2 - Parsing and counting an input sentence by word

Enter 1 or 2 to begin your task or anything else to quit: z

End of Assignment-3

C:\2510>
```

Fig-3

```
C:\2510>java Assignment3

Assignment-3 can perform the following two tasks:
        1 - Making 3-member teams from class of COP2510
        2 - Parsing and counting an input sentence by word

Enter 1 or 2 to begin your task or anything else to quit: 1

Enter total number of teams: 0

Enter 1 or 2 to begin your task or anything else to quit: 1

Enter total number of teams: 1
Team 1:
        Garrett Ford
        Marissa Bondoc
        Evangeline Polepaka

Enter 1 or 2 to begin your task or anything else to quit: 1

Enter total number of teams: 3
Team 1:
        David Debois Sousa
        Andrey Kukushkin
        Raquel Pereda
Team 2:
        Geoffroy De Blegiers
        Stephen Diehl
        Susanna Singleton
Team 3:
        David Hatcher
        Kaylie Johnson
        Yi Tao

Enter 1 or 2 to begin your task or anything else to quit:
```

Fig-4

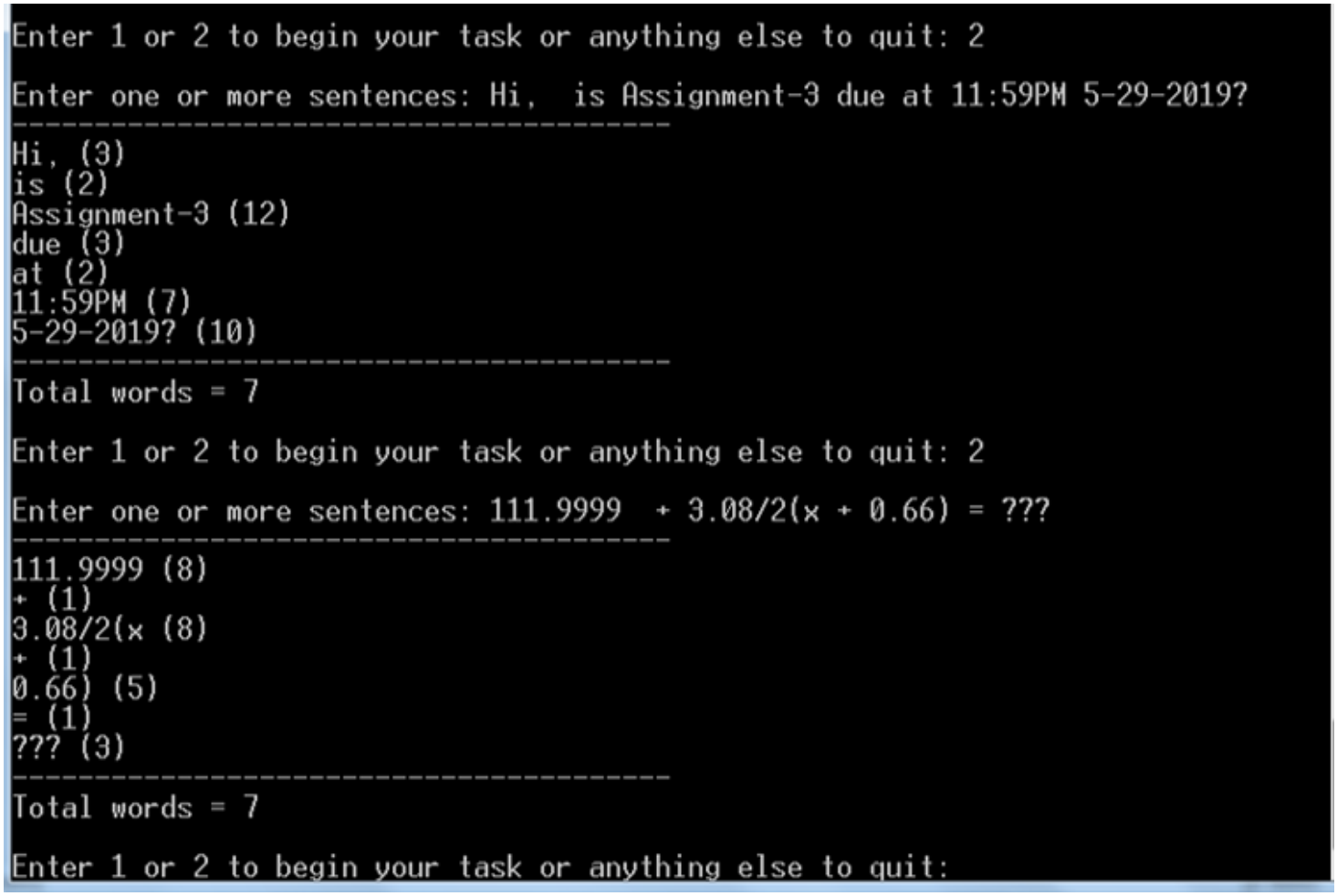
Enter 1 or 2 to begin your task or anything else to quit: 2
Enter one or more sentences: Hi, is Assignment-3 due at 11:59PM 5-29-2019?
Hi, (3)
is (2)
Assignment-3 (12)
due (3)
at (2)
11:59PM (7)
5-29-2019? (10)
Total words = 7
Enter 1 or 2 to begin your task or anything else to quit: 2
Enter one or more sentences: 111.9999 + 3.08/2(x + 0.66) = ???
111.9999 (8)
+ (1)
3.08/2(x (8)
+ (1)
0.66) (5)
= (1)
??? (3)
Total words = 7
Enter 1 or 2 to begin your task or anything else to quit:

Fig-5

```
Enter 1 or 2 to begin your task or anything else to quit: 2

Enter one or more sentences: THIS IS    COP       2501-001    !
---------------------------------------
THIS (4)
IS (2)
COP (3)
2501-001 (8)
! (1)
---------------------------------------
Total words = 5

Enter 1 or 2 to begin your task or anything else to quit: 2

Enter one or more sentences: 1122334455.00009    zhjAU9=dlk$ (773676#
---------------------------------------
1122334455.00009 (16)
zhjAU9=dlk$ (11)
(773676# (8)
---------------------------------------
Total words = 3

Enter 1 or 2 to begin your task or anything else to quit: 2

Enter one or more sentences:
---------------------------------------
Nothing or only space(s) is entered.
---------------------------------------
Total words = 0

Enter 1 or 2 to begin your task or anything else to quit:
```

Operation

- The is a Java console application that can perform two different tasks:
  1. Making one or more teams, each consists of three members who are randomly selected from the 49 students of class COP2510. The text file below contains all 49 names of students in no specific order.
     - **COP2510.txt**
  2. Counting total words of sentence(s) entered from keyboard.
- The application first prompts the above two tasks and asks user to enter 1 or 2 to start the task, or anything else to quit. See Fig-1. This process is repeated every time when a task is completed.
- The application quits with a message "End of Assignment-3". See Fig-2.
- **Team Maker:**
  1. If 1 is entered, the application prompts the user to enter the desired number of teams to make.
  2. If 0 is entered, nothing happens and the application continues on prompting the user to enter 1 or 2. See Fig-3.
  3. If the number of teams is 1 or greater, the application displays the team number, beginning from 1, followed by three full names randomly selected from the 49 students provided in COP2510.txt. See also Fig-3, where 1 and 3 are entered for making one team and three teams, respectively. Hint: Use

the Random class as seen in GetRandom1.java or GR.zip of Quiz 3 to implement this random selection.

4. All names of each team must be displayed beginning with a 'tab'.
5. It's very important in this application that no student appears in the same or different teams more than once.
6. Hint: there are more than one way to "map" a random number (integer) to a specific student name. Using if....else if....else if....else if.... is one possible approach and is recommended here. Storing all names in an array is another way but is not introduced until Chapter 10 .

- **Counting Words:**
  1. If 2 is entered, the application prompts the user to enter one or more sentences. See Fig-4.
  2. The application uses the space character, " ", to separate and count words in the input. If there are two or more consecutive spaces, they are treated as just one. That is, "AA BB" and "AA BB" both contains two words. See Fig-5.
  3. All leading and trailing spaces in the input would be ignored. Hint: use 'trim()' method of String.
  4. The application display two lines of dashes, i.e., "----------" to enclose every word it finds. Each word must be displayed together with its length. For example, if "Hi, John!" is entered, the two lines between dashes should be "Hi, (3)" and "John!(5)".
  5. After the 2nd dashes line, the total number of words is displayed to end the task.
  6. If no words or sentences are entered, a message of "Nothing or only space(s) is entered." is displayed between the two dashes lines and the count is zero. See the last input in Fig-5.
  7. Hint: You may use **trim()**, **indexOf()**, **length()**, **substring()**, and **equals()** methods of String to implement the above word count task. Even the same methods are used, there are different approaches to get this task completed.