Assignment 1: Tinkering with Java

Assignment for COSC 1200 – Object Oriented Programming I

Prior to attempting this problem, you should have done the following:
1. Viewed the content, including assigned readings for weeks 1, 2, 3, and 4.
2. Viewed the lectures and practiced with the demo code for weeks 1, 2, 3, and 4

General Requirements

1. This assignment is to be completed **individually**. Group submissions will not be considered.
2. Analyze the problem, it is recommended (but not required, will not be graded) to develop a plan your solution using a flowchart or pseudo code.
3. Submit your solution to the appropriate assignment folder on DC Connect **by the due date provided**.
4. Your instructor will assign a grade and post feedback on your submission to DC Connect.

Program Requirements

Source code for this assignment can be found in the Assignment 1 folder on DC Connect. Download, make a local copy, and import the provided source code it your assignment project. Complete and submit the program source code that satisfies the following requirements:

1) Create from scratch a Java class named CourseInfo that includes five (5) lines to be outputted (tabbed in once):
    a. The first line will state your full name and program (full program name).
    b. State your professor's name
    c. State the course code and name of the course
    d. State the weekdays and times
    e. State the room numbers tabbed to line up with the appropriate weekday and time

Example output:

```
    <your name here> - Computer Programming & Analysis
    Professor - Jennifer Short
    COSC 1200 Object Oriented Programming I
    Tuesdays 10am - 12pm    Thursdays 5pm - 7pm Fridays 12pm - 2pm
    Room A150               Online              Room C309

Process finished with exit code 0
```

2) Import the source file named VariableDemo.java into your Assignment 1 project. Complete the following:
    a. Compile and run the file to ensure it works
    b. Add code to VariableDemo.java that declares and initialize, with appropriate names and values, primitive variables of type short, long, byte, and float. Remember to qualify your right-hand operand with the appropriate suffix where appropriate.
    c. Add code to display **all** of the newly declared variables. Note: Each variable must have an appropriate label to explain what it is.
    d. Finally, add code that places the contents of your float into your variable type long. Note: This will require that you **cast** the variable

3) Create a java class tile named IfDemo.java. This class should satisfy the following requirements:
    a. Create appropriately named primitive variables, one will be a character (char type) used to store a letter grade, one to store an integer (int type) variable used to store a final grade.
    b. It will also have an if… else statement that uses a final grade score to assign the letter grade using the following chart:

Final Grade	Letter Grade
90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
50 – 59	E
< 50	F

    c. After the if…else statements, create a String object named status, use the conditional operator to initialize this object based on if the letter grade if ‘F’ they “failed” otherwise, they “passed”. Also display the letter grade.

4) Create a java class file named SwitchDemo.java. This class should satisfy the following requirements:
    a. Declare and initialize a character variable with a letter grade.
    b. Has nested if…else statements that performs the following checks:
        i. If the letter grade is 'A', a message stating "Great work" is displayed
        ii. If the letter grade is 'B', a message stating "Good work" is displayed
        iii. If the letter grade is 'C', a message stating "Average work" is displayed
        iv. If the letter grade is 'D', a message stating "Below average work" is displayed
        v. If the letter grade is 'F', a message stating "Unsatisfactory work" is displayed
        vi. If the letter is anything other than the above values, a message stating that the letter was invalid should be displayed
    c. Once the nested if…else is completed, replicate the functionality using a switch statement instead

5) Import the file LoopDemo.java provided into your Assignment 1 project. Execute it to ensure it works. Add the following functionality:
    a. Create two nested for loops, the outer loop should start at 1 and should stop executing when its counter gets to 3. The inner loop should start at 0, and stop executing when the counter is greater than 2. The inner loop should display both the outer and inner counters.
    b. Create a do…while loop that starts at 0 and counts up to 10 by 2s. The loop should display the counter each time through.
    c. Create a while loop that starts at 9 and counts down by 3s, and stops executing when the counter is less than or equal to zero. The counter should be displayed each time.

Style Guide Notes

To be eligible for full marks on this or any assignment in this course your application must conform to the requirements as outlined above as well as the Google style guide as well as appropriate and complete program documentation.