Programming Exercise 01 - Basic Java

Authors: Rachna, Nathaniel, David, Faris, Nicholas, Julia

Problem Description

This assignment will test your basic knowledge of variables, expressions, assignment, classes, and String output.

There are two parts:

1. Calculating the tip after your meal
2. Decide on how much fuel per person for a car trip

Solution Description

Part 1: Tip after getting total for a meal

1. Create a class called `TipHelper`
2. Add a block comment at the top that includes your name and one thing you wish more people knew about you (have fun with it!)
3. Inside `TipHelper`, create the main method
4. Within the main method:
5. Create a variable named `tipPercentage` and assign it the value 0.15. This variable should hold floating point numbers and should be the larger of the two types of floating point variables.
6. Create a variable named `mealTotal` and assign it the value 20.17. This variable should hold floating point numbers and should be the larger of the two types of floating point variables.
7. Create a variable named `numberItemsOrdered` and assign it the value of 3. This should be an int because you can only order a whole number of items.
8. Create a variable named `customerName` and assign it the value "Jacob". This should be a String.
9. Next, calculate the tip and store it in a variable called `totalTip` using a single expression. This variable should be the same type as `mealTotal`.
   (Remember, `totalTip = tipPercentage * mealTotal`)
10. Then, calculate the total amount owed with tip and store it in a variable called `totalAmountPaid` in a single expression. This variable should also be the same type as `mealTotal`.
    (Remember, `totalAmountPaid = totalTip + mealTotal`)
11. Finally, copy paste the following statements into your code to print out the final values. **Make sure to use those exact statements to get the right decimal precision**. (We'll cover printf in a future lecture.)

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

```
System.out.printf("The total tip was $%.2f!\n",
totalTip);

System.out.printf(customerName + " owes a total of
$%.2f.\n", totalAmountPaid);
```

12. Here is what the output should look like for the given values:

```
The total tip was $3.03!

Jacob owes a total of $23.20.
```

Part 2: Calculate Fuel Needed

1. Create a class called `FuelTripPlanner`
2. Inside `FuelTripPlanner`, create the main method
3. Inside the main method:
4. Create an int variable named `backRoadMiles` and assign it 25
5. Create an int variable named `highwayMiles` and assign it 60
6. Create an int variable named `hillyMiles` and assign it 10
7. Create an int variable named `currentGasGallons` and assign it 20
8. Finally, create an int variable named `numberOfPeople` and assign it to 5
9. Create a variable of type double called `gallonsPerPerson` and initialize it to 0.
10. Using Compound Assignment operators (+=, -=, etc), perform the following calculations to `gallonsPerPerson`.
    a. Add `backRoadMiles` multiplied by 2 (meaning 2 gal/mile on back roads)
    b. Add `highwayMiles` multiplied by 1 (meaning 1 gal/mile on highways)
    c. Add `hillyMiles` multiplied by 5 (meaning 5 gal/mile on hilly roads)
    d. Subtract `currentGasGallons` to account for the gas already in the car
    e. Divide by `numberOfPeople` (Watch out for integer division!)
11. Copy this exact statement into your code to print out the final value of gallonsPerPerson. (We'll cover using printf to get the right number of decimals in a future lecture).

```
System.out.printf("Each passenger is responsible for %.1f
gallons of gas.\n", gallonsPerPerson);
```

12. Here is what the output should look like with the given numbers:

```
Each passenger is responsible for 28.0 gallons of gas.
```

Turn-In Procedure

Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- `TipHelper.java`
- `FuelTripPlanner.java`

Make sure you see the message stating "PE01 submitted successfully". From this point, Gradescope will run a basic autograder on your submission as discussed in the next section. **Any autograder test are provided as a courtesy to help "sanity test" your work and you may not see all the test cases used to grade your work.** You are responsible for thoroughly testing your submission on your own to ensure you have fulfilled the requirements of this assignment. If you have questions about the requirements given, reach out to a TA or professor via the class forum for clarification.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the assignment (submit early and often). We will only grade your latest submission: be sure to **submit every file each time you resubmit**.

Gradescope Autograder

For each submission, you will be able to see the results of a few basic test cases on your code. If you fail a test, you can look at the output to determine what went wrong and resubmit once you have fixed the issue.

The Gradescope tests serve two main purposes:

1. Prevent upload mistakes (e.g. forgetting checkstyle, non-compiling code)
2. Provide basic formatting and usage validation

In other words, the test cases on Gradescope are by no means comprehensive. Be sure to thoroughly test your code by considering edge cases and writing your own test files. You also should avoid using Gradescope to compile, run, or checkstyle your code; you can do that locally on your machine. (We are not using checkstyle for this programming exercise, but in future HWs we will).

Other portions of your assignment can also be graded by a TA once the submission deadline has passed, so the output on Gradescope may not necessarily reflect your grade for the assignment.

Allowed Imports

To prevent trivialization of the assignment, you are not allowed to use any imports.

Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach or break our auto grader. For that reason, do not use any of the following in your final submission:

- `var` (the reserved keyword)
- `System.exit`

Collaboration

Only discussion of the assignment at a conceptual high level is allowed. You can discuss course concepts and HW assignments broadly, that is, at a conceptual level to increase your understanding. If you find yourself dropping to a level where specific Java code is being discussed, that is going too far. Those discussions should be reserved for the instructor and TAs. To be clear, you should never exchange code related to an assignment with anyone other than the instructor and TAs.

You **MAY NOT** use code generation tools to complete this assignment. This includes generative AI tools like ChatGPT.

Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items
- Do not submit `.class` files
- Test your code in addition to the basic checks on Gradescope
- Submit every file each time you resubmit
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points
- **Check on Ed Discussion for a note containing all official clarifications and sample outputs**

It is expected that everyone will follow the Student-Faculty Expectations document, and the Student Code of Conduct. The professor expects a **positive, respectful, and engaged academic environment** inside the classroom, outside the classroom, in all electronic communications, on all file submissions, and on any document submitted throughout the duration of the course. **No inappropriate language is to be used, and any assignment, deemed by the professor, to contain inappropriate, offensive language or threats will get a zero**. You are to use professionalism in your work. Violations of this conduct policy will be turned over to the Office of Student Integrity for misconduct.

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.