Programming Exercise 02

Authors: Ian, Brittney, Andrew, Aanya, Vince, Allison

Problem Description

This assignment will test your basic knowledge of Type Conversion, Strings, and Command Line Input/Output. In this PE you will exercise ...You will be writing 2 programs for the following scenarios: building an elevator pitch and a basic four-function calculator.

Solution Description

Creating an Elevator Pitch

You and your friends are at the career fair and need help creating a simple elevator pitch. As the programmer of the group, your friend's future job prospects lie in your hands. Are you up for the challenge?

1. Create a class called `ElevatorPitch` and create a `main` method inside it.
2. Inside the main method do the following:
   - Create a variable called `myPitch` that will save your completed elevator pitch as a String.
   - Create a variable called `name` that will save your full name as a String.
   - Create a variable called `gpa` that will save your GPA (floating point number).
   - Create a variable called `graduationYear` that will save your graduation year (integer).
   - Create a variable called `hobby` that will save your favorite hobby as a String.
   - Create a variable called `major` that will save your current major as a String.
3. Declare a Scanner variable called myScanner that will take in input from the user. Create a Scanner object and assign it to the myScanner variable. Use this Scanner object to read user input for the following steps. **Do NOT instantiate multiple `Scanners`** in `ElevatorPitch`.
4. Print to the console "`Enter your full name: `"
5. Read the value inputted by the user and assign it to `name`. You should use `nextLine()` to do this.
6. Print to the console "`Enter your gpa: `"
7. Read the value inputted by the user and assign it to `gpa`. You should use `nextDouble()` to do this. Note that `nextDouble()` leaves a newline character in your terminal (from when you pressed enter), which you will need to get rid of with an extra `nextLine()` call.
8. Print to the console "`Enter the year you will graduate: `"
9. Read the value inputted by the user and assign it to `graduationYear`. You should use `nextInt()` to do this. Note that `nextInt()` leaves a newline character in your terminal (from when you pressed Enter), which you will need to get rid of with a call to `nextLine()`.
10. Print to the console "`Enter your favorite hobby: `"
11. Read the value inputted by the user and assign it to `hobby`. You should use `nextLine()` to do this.
12. Print to the console "`Enter your major: `"
13. Read the value inputted by the user and assign it to `major`. You should use `nextLine()` to do this.
14. Assign the following String to `myPitch`:
    "`Hello, my name is {name}. I'm a {major} major with a gpa of {gpa} graduating in {graduationYear}. In my free time, I like {hobby}.`"

Where {*name*},{*major*},{*gpa*},{*graduationYear*}, and {*hobby*} are the values of the corresponding variables without curly braces.

15. Print `myPitch` to the console.

NOTE: Prompts, inputs, and output should each be printed on separate lines. You can assume only integers will be inputted for graduation year and only integers or floating point numbers will be inputted for GPA.

Example input and output (we may test other values):

```
Enter your full name:
Dobby Bodd
Enter your gpa:
-1.2
Enter the year you will graduate:
3005
Enter your favorite hobby:
getting tested
Enter your major:
computer science
Hello, my name is Dobby Bodd. I'm a computer science major with a gpa
of -1.2 graduating in 3005. In my free time, I like getting tested.
```

Creating a Four-Function Calculator

You show up to your first calculus course at Georgia Tech and your calculator is out of batteries. Luckily, you only need four basic operations for your first day: addition, subtraction, multiplication, and division.

1. Create a class called `Calculator` and create a main method inside it.
2. Create a variable called `firstNumber` which will always be an integer.
3. Create a variable called `secondNumber` which will always be an integer.
4. Declare a `Scanner` called `myScanner` that will take in input from the user. **Do NOT instantiate multiple `Scanners` in `Calculator`.**
5. Print to the console "`Enter your first number: `"
6. Read the value inputted by the user and assign it to `firstNumber`. You should use `nextInt()` to do this.
7. Print to the console "`Enter your second number: `"
8. Read the value inputted by the user and assign it to `secondNumber`. You should use `nextInt()` to do this.
9. Print out "`Sum is `" followed by the sum of the two numbers (`firstNumber` + `secondNumber`)
10. Print out "`Difference is `" followed by the difference of the two numbers (`firstNumber` – `secondNumber`)
11. Print out "`Product is `" followed by the product of the two numbers (`firstNumber` * `secondNumber`)
12. Print out "`Quotient is `" followed by the quotient of the two numbers (`firstNumber` / `secondNumber`).
    The result should be displayed as a decimal regardless of whether the quotient is a real or number (e.g. print `2.5` for `5/2 and p`rint `4.0` for `4/2`). In order to do this, you will need to use casting to force floating point division.

NOTE: Prompts, inputs, and output should each be printed on separate lines. You can assume only integers will be inputted for the first and second number.

Example input and output (we may test other values):

```
Enter your first number:
3
Enter your second number:
5
Sum is 8
Difference is -2
Product is 15
Quotient is 0.6
```

Submitting

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

- `ElevatorPitch.java`
- `Calculator.java`

Make sure you see the message stating "PE01 submitted successfully." From this point, Gradescope will run an autograder on your submission verifying all necessary files are included and working as intended. Note: the nature of some assignments means they are **not fully autogradable** and will require some manual grading after submission. Although all autograder tests will be made visible on this assignment, this is **NOT always** your final grade. You can submit as many times as you want before the deadline, so feel free to resubmit as you make progress on the homework. We will only grade your last submission be sure to **submit *every file* each time you resubmit.**

Import Restrictions

You may not import anything for this homework assignment **except for** `java.util.Scanner`

Feature Restrictions

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

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

Collaboration

No collaboration is allowed on this assignment. See syllabus for more details.

In addition, note that you are not allowed to upload your code to any sort of public repository. This could be considered an Honor Code violation, even if it is after the homework is due.

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 Piazza for all official clarifications