CSE 8B Spring 2022

Assignment 2

Selections, Mathematical Operations, Strings, Characters, Loops and Methods in Java

Due Date: Wednesday, April 13, 11:59 PM

Learning goals:

- Write Java code to build classes that include:
    - Selections
    - Mathematical operations
    - Strings
    - Loops
    - Methods
    - Scanner
- Write your own test cases to test the correctness of your methods

NOTE: This programming assignment must be done individually. Paired programming is NOT allowed for this assignment.

Coding Style (10 points)

For this programming assignment, we will be enforcing the CSE 8B Coding Style Guidelines. These guidelines can also be found on Canvas. Please ensure to have COMPLETE file headers, class headers, and method headers, to use descriptive variable names and proper indentation, and to avoid using magic numbers.

Part 0: Getting started with the starter code (0 points)

1. Make sure there is no problem with your Java coding environment. If there is any, then review Assignment 1, or come to the office/lab hours before you start Assignment 2.
2. Unlike the previous PA, in this PA, you will start with some code that we have written for you.

3. If you are using UCSD Linux Cloud (recommended) for the assignment, download the starter code from Piazza → Resources → Homework → Assignment2.java by opening a web browser on the Remote Desktop. You can open Firefox by going to Applications → Firefox.
4. If you are working on your local machine, then you can download the starter code from a local web browser on your machine.

Part 1: Implement methods (50 points)

In `Assignment2.java`, you will be calculating a statistic about COVID-19 cases in two different approaches and drawing a solid circle pattern similar to the COVID-19 virus in Java. **You will be writing a total of 3 methods.**

Before you implement your methods, please take a look at the constants at the top of the class. All of the error messages as well as method-specific variables are already defined for you with the keywords `private final static`. **IMPORTANT: You should use these constants when developing your methods to ensure that your code will always give the correct output.**

You are provided a mostly empty `Assignment2` class in the file `Assignment2.java`. The below methods have already been declared in the class. Complete the methods as specified below.

Methods to be implemented

In this assignment, we are going to implement two methods that compute the total number of COVID-19 cases on the n^{th} day given the number of cases on the 1^{st} day and the assumption that the number of cases compounds every day by a constant rate, i.e., the number of cases on $(x+1)^{th}$ day increases by a constant percentage compared to the x^{th} day. We implement a third method that draws a solid circle pattern using for loops that resembles a COVID-19 virus.

1. `public static int findNumCasesUsingLoop(int initial, double rate, int n)`

   This method should return the total number of cases/infections (as an `int`) on the n^{th} day given the total number of cases on the 1^{st} day and the rate of increase, using **a for loop**. The total number of cases computed would be a `double` value but should be returned as an `int` by rounding to the nearest integer not less than the `double` value and type casting to an `int`.

2. `public static int findNumCasesUsingPower(int initial, double rate, int n)`

This method should return the total number of cases/infections (as an `int`) on the n^{th} day given the total number of cases on the 1^{st} day and the rate of increase, **using the below formula (similar to compound interest) with Math.pow()**:

$$nthDayCases = initialCases * (1 + rate)^{n}$$

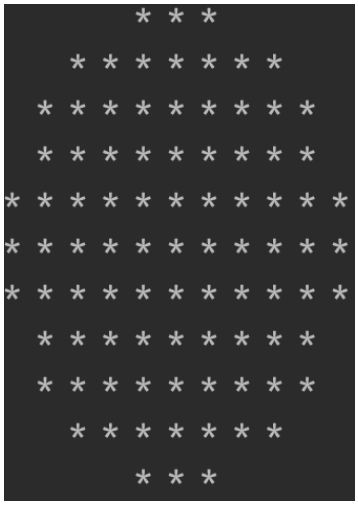
The total number of cases computed would be a `double` value but should be returned as an `int` by rounding to the nearest integer not less than the `double` value and type casting to an `int`.

3. `public static String drawCovidVirus(char ch, int rad)`

This method should draw the COVID-19 virus using the `ch` character with the given `rad` and return the resulting pattern as a `String`. Don't worry, we won't be actually drawing the COVID-19 virus but just a solid circle pattern with the given character and radius. We use two nested for loops to draw the solid circle pattern. We assume the center of the circle to be at origin `(0,0)` and compute the x and y coordinates of points for each x in range `(-rad, rad)` and y in range `(-rad, rad)` using two nested for loops . For each point (x, y), we need to check if the point lies inside or on the circle using the circle equation:

$$point = x^2 + y^2 - rad^2 - 1$$

If `point` is less than or equal to 0, then the `point` lies inside or on the circle and we append the character `ch` followed by a ' ' (ch followed by a single space), else we append only ' ' (two spaces) to the output string. The output string is returned by the method which is printed to console in the main method. For example, when `ch` is '*' and `rad = 5`, the output string should be as follows:



NOTE: You can assume that the character will always be a valid character.

Part 2: Compile, Run and UnitTest Your Code (10 points)

Testing is a very important part in programming. In this course, we will get you familiar with unit testing. For this assignment and all future assignments, you will be asked to create your own tests to check whether your code works as expected. **In this part of the assignment, you need to implement your own test cases in the method called `unitTests`.**

In the starter code, some test cases are already implemented for you. You can regard it as an example to implement other cases. The general approach is to come up with different inputs and manually give the expected output, then call the method with that input and compare the result with expected output.

You are encouraged to create as many test cases as you think to be necessary to cover all the edge cases. The `unitTests` method should `return true` only when all the test cases are passed. Otherwise, it should `return false`. **To get full credit for this section, you should create at least five test cases that cover different situations (including the ones we have provided) for the first two methods.** In other words, you will need to create at least **three** more tests that test both the methods `findNumCasesUsingLoop` and `findNumCasesUsingPower.` Note that you need to test both the methods for each test case similar to the ones already provided. We skip the unit testing of the `drawCovidVirus` here due to the complexity involved in creating expected outputs.

Part 3: Complete `main` (30 points)

After completing the three methods and creating several unit tests, compile and run `Assignment2`. You should see the message **"All unit tests passed"**. If not, then it is very likely that you have bugs in your code. Read the previous parts carefully while inspecting your code to fix your bugs. We call this process debugging. The main method is the method that will be called when running program `Assignment2`. Here is the main method given to you in the

starter method:

```
// TODO: Complete the method body
Run | Debug
public static void main(String[] args) {
 // Perform unitTests first
 if (unitTests()) {
     System.out.println("All unit tests passed.\n");
 } else {
     System.out.println("ERROR: Failed test.\n");
     return;
 }

 // Start the user-machine interaction
 // TODO: Initialize Scanner object

 // Continuously loop until the user inputs "end"
 while (true) {
     // TODO: Complete user-machine interaction here

     break; // TODO: Remove this break when you complete user-machine interaction
 }

 // TODO: Don't forget to close the scanner at the end of the program
 }
```

The code to run `unitTests` is already given to you. **Do NOT change any code above the comment `"// Start the user-machine interaction"`.**

Below that comment, your final task is to implement an *ask-answer* interaction functionality via command line.

First, your program should print the prompt `"Which method do you want to call? Type \"end\" to stop the program."` and wait for the user to enter feedback. **Remember to use the constants defined at the top of the class for all the prompts displayed in the main method.** After the method name is entered, it will read the input method name via Scanner. Then:

1. If the input method name is `"findNumCasesUsingLoop"`, the program should take the following inputs from the user using Scanner class displaying the corresponding prompts before taking input.
    - Initial number of cases for state 1
    - Rate of increase for state 1
    - Initial number of cases for state 2

- Rate of increase for state 2
- Value of 'n'

Invoke the method `findNumCasesUsingLoop` for state 1 and state 2 and compute the total number of cases on the n^{th} day for both the states and print the state with the higher number of n^{th} day cases.

2. If the input method name is "`findNumCasesUsingPower`", the program should take the following inputs from the user using Scanner class displaying the corresponding prompts before taking input.
    - Initial number of cases for state 1
    - Rate of increase for state 1
    - Initial number of cases for state 2
    - Rate of increase for state 2
    - Value of 'n'

    Invoke the method `findNumCasesUsingPower` for state 1 and state 2 and compute the total number of cases on the n^{th} day for both the states and print the state with the higher number of n^{th} day cases.

3. If the input method name is "`drawCovidVirus`", the program should print the prompt `"Please enter a character."`. It will then read the user input as a character variable. As noted before, you can assume that the character variable will always be a valid character. Then, the program should print another prompt `"Please enter a radius (as an integer)."` and read the user input as an integer variable. Then, call method `drawCovidVirus` with argument `ch` and `rad` which returns the pattern as a `String`. Finally, the program will print out the result.

4. If the input is `"end"`, return from the main method by breaking out of the infinite loop, which will terminate the execution of the program.

5. If the input is invalid (e.g., when input method name is misspelled), print the error message: `"Invalid method - options are: findNumCasesUsingLoop, findNumCasesUsingPower and drawCovidVirus."`

After performing one of the above, your program should print the prompt `"Which method do you want to call? Type \"end\" to stop the program."` again and repeat the whole process. The repetition should never stop until `"end"` is entered.

Example: you should be able to *exactly* reproduce this output below with your program.

```
(base) ➜  Documents javac Assignment2.java
(base) ➜  Documents java Assignment2
All unit tests passed.

Which method do you want to call? Type "end" to stop the program.
findNumCasesUsingLoop
Enter initial number of cases in state 1:
50
Enter rate of increase for state 1:
0.2
Enter initial number of cases in state 2:
30
Enter rate of increase for state 2
0.3
Enter day n:
16
Number of cases on day 16 in state 1: 925
Number of cases on day 16 in state 2: 1997
State 2 has more cases on day 16

Which method do you want to call? Type "end" to stop the program.
findNumCasesUsingPower
Enter initial number of cases in state 1:
50
Enter rate of increase for state 1:
0.2
Enter initial number of cases in state 2:
30
Enter rate of increase for state 2
0.3
Enter day n:
16
Number of cases on day 16 in state 1: 925
Number of cases on day 16 in state 2: 1997
```

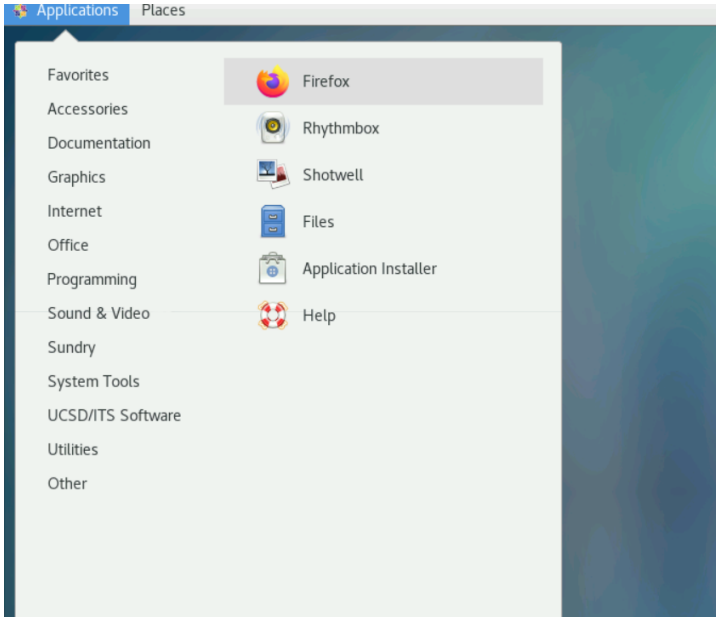
```
State 2 has more cases on day 16

Which method do you want to call? Type "end" to stop the program.
foo
Invalid method - options are: findNumCasesUsingLoop, findNumCasesUsingPower and drawCovidVirus

Which method do you want to call? Type "end" to stop the program.
drawCovidVirus
Please enter a character.
*
Please enter a radius (as an integer).
5
        * * *
    * * * * * * *
  * * * * * * * * *
  * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
  * * * * * * * * *
  * * * * * * * * *
    * * * * * * *
        * * *

Which method do you want to call? Type "end" to stop the program.
end
```

NOTE: Notice that there is an empty line after each method output. It is very important that your program also prints this empty line!

Submission

You're almost there! Please follow the instructions below carefully and use the **exact submission format**. Because we will use scripts to grade, **you may receive a zero** if you do not follow the same submission format.

1. Open a web browser on the Remote Desktop. You can open Firefox by going to Applications → Firefox.



2. Open Gradescope in Firefox and login. Then, select this course → PA2.
3. Click the DRAG & DROP section and directly select the required file **Assignment2.java**. Drag & drop is fine. Please make sure you don't submit a zip, just the file in one Gradescope submission. Make sure the name of the file is correct.
4. You can resubmit unlimited times before the due date. Your score will depend on your final (most recent) submission, even if your former submissions have higher scores.
5. Your submission should look like the below screenshot. If you have any questions, feel free to post on Piazza!