

## Small Assignment #1

**Due: Wednesday, 1/29/2025 by 11:59 PM**

**Submission:** Submit this to Gradescope as a single PDF. Assign pages to questions. Points will be deducted if you do not assign pages correctly.

### Question 1. (5 points)

In the Software Development Tools slides, there are three tasks you are asked to complete.

- (a) Indicate here that you have successfully installed git on your computer. Just answer YES or NO.
- (b) Indicate here that you have successfully made an account on github by providing the email address/username associated with your account.
- (c) Indicate here that you have successfully completed the three training courses on github that are listed in the slides under Task #3. Providing a screenshot showing completion for each of the courses is sufficient.

### Question 2. (5 points)

Code is provided for this question. Find all the compile-time errors in this code and explain why each of them causes the code to not compile. Remember that the compiler is primarily focused on analyzing the *source code*, so your explanations should clearly reference that.

### Question 3. (8 points)

Code is provided for this question. NOTE: Do not run the code right away. Follow the instructions carefully and answer these questions *in order*. Many of them are graded for completion because the process here is important for thinking through the concepts, so *do not go back and change your answers after running the code or answering later questions*.

- (a) Look at the code carefully. In this question, put your prediction for what is printed out when Main is run.
- (b) Now run the code and check your prediction. Were you wrong about any of them?
- (c) Take some time to explain each part of this to yourself. Remember that you may be asked these kinds of questions on quizzes and exams, so it is in your best interest to make sure you can do it. Use appropriate terminology from class.
- (d) In your opinion, is Student.java a well-designed class? Explain your answer.
- (e) In your opinion, is StudentList.java a well-designed class? Explain your answer.
- (f) Does age\_bob++ actually change Bob's age? Explain why or why not.
- (g) One of the fundamental principles of object-oriented programming is *encapsulation*. To put this idea succinctly, the rule of encapsulation is:

*Code from outside a class should not be able to change a class's internal data without going through the class's public methods.*

Do Student.java and StudentList.java exhibit good encapsulation? Explain your answer.