**CS 113 & CS 109**

**Fall 2023** Final

Instructions and Rules (These have changed slightly)

1. You will receive an email from me indicating that the exam is available no later than 9am Sunday Dec 17. You may begin the exam any time after you receive that email. You must complete the exam prior to the end of the exam period, Friday Dec 16 at 12:30pm.
2. You have **200** minutes from the time you hit the button to obtain the exam until you return it to me. Included in this time limit is the time to: obtain the exam; return the exam; have a printer failure, have other electronic mishaps; etc. If something truly dire happens during your 200 minutes, email me immediately.
3. On the first page of your submission clearly write the time you started and the time you completed the exam. Also sign by that time to indicate that you have abided by the honor code. I have a record of the time you obtained the exam and a timestamp of when you emailed it back to me. Still, note these times requested above on your submission.
4. All answers should be written on blank (or lined) sheets of paper. Also, answers may be typed or written using a tablet or computer. I have left lots of blank space if you take the tablet writing approach.
5. SUBMISSION: send your answers to gtowell@brynmawr.edu. If you typed everything, or wrote on a tablet, just copy and paste or attach the document. For handwritten answers, take photos or scan. Please ensure legibility. You submission should be in PDF or some other common format. Your submission should be an actual file, NOT a link to a file on onedrive, dropbox, etc.
6. This is an open everything exam. *There are two restrictions: first, you may not discuss an aspect of the exam with anyone. The only discussion of the exam that would not violate the honor code would be "I have not taken it*

*yet". Even commenting that you thought the exam was hard (or easy) would be a violation. Second, you may not use an AI service (eg ChatGPT).*

7. In order to be eligible for as much partial credit as possible, show all of your work for each problem, and clearly indicate your answers. Credit will not be given for illegible answers or for answers that I cannot find.

8. Code that you write should be as close to correct (runnable) as possible. Small syntax errors will not cost you points, but code that is unclear will. You may write your code using VSC; you may compile and test. I do not encourage this (I even discourage doing so as it will tend to slow you down), I just note that it is permitted.

9. If you are totally blocked on writing code (on questions that call for writing code) you may write pseudo code — or even english — describing as precisely as possible what the code should do. Such an answer will not receive full credit, but it will receive some credit.

10. If you make an assumption (or are unsure of your interpretation of a question), state that assumption or interpretation clearly. For instance you might say "I assume that the question means that I should do …". If I agree that your interpretation is valid, then your answer will be eligible for full credit. However, I may not agree with your interpretation or I may feel that you interpretation is only worth partial credit.

11. While you are taking the exam, you can email me with questions. However, I make no promises about when I will be available. Generally, you are better off using point 10.

12. You may use any function defined in Java 11. BUT if you use functions not discussed in class or in the text I will worry. See the next point.

13. *You are responsible for fully understanding your responses. For instance, if you read on stack overflow "just multiply by 2.3" you should know why you are multiplying by 2.3. If, for some reason, I do not believe you understand your answer, I may take points off even if your answer is correct. So if you do something weird (ie not in the book or discussed in class) then provide an explanation sufficient to prove your understanding.*

```
public class Fund {
    private String name;
    private int goal;
    private int pledged;

    public Fund(String na, int gl) {
        goal = gl;
        name = na;
        pledged = 0;
    }
    public void donate(int amt) {
        if (amt > 0) {
            pledged += amt;
        }
    }
    public String getName() {
        return name;
    }
    public int getGoal() {
        return goal;
    }
    public int getPledged() {
        return pledged;
    }
    public String toString() {
       return name + " has a goal of " + goal + " and is pledged " +
pledged;
    }
}
```

---

```
public class FundGroup {
    private Fund[] collectionOfFunds;
    private int knownFunds;

    public FundGroup() {
        collectionOfFunds = new Fund[10];
        knownFunds = 0;
    }

    public void addFund(String fundName, int fundGoal) {
        Fund f = new Fund(fundName, fundGoal);
        collectionOfFunds[knownFunds] = f;
        knownFunds += 1;
    }

}
```