COMP 50 Final exam: the learning portfolio

Most courses have a traditional final examination in which the instructor chooses a subset of course topics and asks you questions. In this course, *you* choose the topic. You are to prepare and submit a *learning portfolio* containing the following elements:

- A sample of your early programming work, either from early in the class or from before class started
- A sample of your work that demonstrates something important you've learned in the class
- A short, reflective essay that explains what you've learned, using your two samples as evidence

The portfolio should *show us that you have learned one important thing*. You pick the thing and argue why it's important. Think of your intended audience as someone who might want to hire you for your problem-solving and program-design skills.

The top strategy for your essay is to choose something that you have *mastered* and that is *important*. It would be bad strategy to try to show that you have mastered many things, or to try to talk about something advanced that you haven't quite mastered. An exemplary essay is narrow and deep, not broad or shallow.

To help you prepare your portfolio, you will have two opportunities to meet with mentors. In November, you'll meet with a subject-matter expert to brainstorm ideas and plan your portfolio. In December, you'll meet with a writing expert who will help you make your portfolio as clear and convincing as possible.

Your portfolio will be evaluated by a group of three evaluators among whom at least one will be a member of the CS faculty. The group will read your portfolio, and then you will meet for 15 minutes: you will have 5 minutes to present your portfolio, and then 10 minutes to answer questions about it. (It is possible that additional members of the faculty may sit in on your presentation, but only the designated evaluators will ask questions.)

The purpose of the oral presentation is to enable you to focus the evaluators' attention on the most valuable parts of your portfolio. The purpose of the question period is to enable the evaluators to clarify anything they may have failed to understand. In both cases my goal is to enable the evaluators to give you the maximum possible credit for the work you will have done.

To help both you and the evaluators to focus, we limit the size as well as the scope. We will accept work samples of up to 100 lines long and essays of up to 4 pages (not including samples in the page count), but you should aim to provide samples of at most 50 lines plus a two-page essay.

If you want to know more about the theory and practice of learning portfolios, I recommend a short position paper by John Zubizarreta. **Be cautioned** that the portfolios discussed in that paper are *far* larger in scope than a COMP 50 portfolio.

Schedule for the portfolio

Your learning portfolio comes with a sequence of activities, some of which must be completed on deadlines:

- Your initial meeting with a mentor should take place by 11:59PM on **Monday, November 18**. Sign up at `http://www.cs.tufts.edu/~nr/cgi-bin/1on1.cgi`.

  Arrive at the meeting with three things:

  - At least three samples of work that might be suitable for inclusion in your portfolio
  - An idea of what learning you might present in your portfolio
  - A backup idea in case the main idea doesn't work out

  This meeting is for *brainstorming*, and it's quite possible that you and your mentor will emerge from this meeting with samples and ideas that are completely different than what you came in with. Your mission is to arrive with samples and ideas to *start* from.

  If you miss this deadline or don't hold this meeting, the grade for your portfolio will be reduced by 15%. If you come to the meeting completely unprepared (no samples and no ideas), the grade for your portfolio will be reduced by 10%. If you come to the meeting partially unprepared, the grade for your portfolio will be reduced by 5%.

  If you feel that your first meeting does not give you a good place to start from, you may request a followup meeting with the professor.

- By **Tuesday, November 26** you must *know the date and time* for your final meeting with a graduate writing consultant (see below).

- You will use the results of your first meeting to create a **complete draft** if your portfolio. You will have about two weeks for this task.

- You will review your *complete* draft with a graduate writing consultant from the Writing Center by **Friday, December 6** at 5:59PM. The consultant will help you make your case as clear and convincing as possible. If you miss this deadline or don't hold this meeting, your grade is reduced by 15%. If your draft is incomplete, your grade is reduced by 10%.

  After your meeting, if you want more help, you can schedule a meeting with a teaching assistant or the professor.

- Your portfolio will be submitted by **Tuesday, December 10 at 11:59PM**. Missing this deadline results in a zero, and there is a chance you will fail the course.

  To submit your portfolio, create file `portfolio.pdf` and copy it to a server such as `linux.cs.tufts.edu`. You can use Word and export PDF, or you can use `pandoc` or any other suitable tool, as long as it ultimately produces PDF.

  Once your file has been copied, submit it using `provide`, as in

  ```
  provide comp50 portfolio portfolio.pdf
  ```

  The skills you will need are the same skills from the lab on Linux and `provide`.

- You will give a 5-minute oral presentation of your portfolio and then spend 10 minutes answering questions, for a total of 15 minutes. The evaluation will be scheduled **during final-exam period**. You will be assigned a 30-minute time slot; your 15 minutes will take place sometime within this slot. If you are not present when called for, your grade will be reduced by 25%. If you are not present by the time your time slot ends, your grade will be reduced by 50%.

  If you want, you can get help with public speaking from the Academic Resource Center. This is an opportunity, not an obligation, but if you want it, book early.

Reductions in grade are *cumulative*, so for example, if you miss both mentor meetings, your grade for your portfolio is reduced by 25%.

What to submit

Email *one* file in PDF format to `nr@cs.tufts.edu`. This file must contain your essay (up to four pages' worth) *plus* your two samples.

- Only files in PDF format will be accepted.
- Samples submitted separately will **not** be accepted.

I recommend that you prepare your PDF using a tool called `pandoc`, which accepts a simple text format and turns it into good-looking text. Pandoc's `--listings` option will produce samples with numbered lines, which we will need in order to be able to discuss your samples.

How your portfolio will be evaluated

Your portfolio will be evaluated on several dimensions. This table shows criteria.

	Exemplary	**Satisfactory**	**Must Improve**
Form	• 2 work samples, at most 50 lines each (the sweet spot is probably 15 to 30 lines) • One work sample is from late in the course. • Up to 4 pages of reflective commentary and analysis. • The essay comes first; the work samples follow as Sample A and Sample B. • The work samples use numbered lines.	• There is a work sample in the range 50–80 lines. • The early work sample is from the course; the late work sample is a new revision of the early one.	• There is a work sample of over 80 lines. • Both samples are from late in the course. • One or both samples are perfunctory or missing (**serious fault**).

	Exemplary	**Satisfactory**	**Must Improve**
Topic	• The topic is central to a course concept (problem-solving by computer) • *Or* the topic is central to the book. • The essay argues explicitly for the importance of what was learned, and the evaluators are convinced by the argument. • The essay emphasizes things the student can *do*. • The essay demonstrates mastery of *technique*. • The learning described in the essay encompasses both data and code.	• The topic is mentioned in book. • The essay argues explicitly for the importance of what was learned, and the evaluators agree that the argument has some merit, but they are not completely convinced. • The essay emphasizes facts or ideas the student *knows*. • The essay demonstrates mastery of *ideas*. • The learning described in the essay is primarily about data. Code is a second-class participant or is not mentioned at all.	• The topic is something that the evaluators believe it not representative of the significant learning expected over the course of a semester. (Example: "Every function should have test cases." This idea is definitely important and central, but by itself it does not represent much accomplishement.) • Neither the evaluators nor the student can connect the topic to the course. • The essay argues explicitly for the importance of what was learned, but the evaluators find the argument without merit. • The essay does not address the issue of the importance of what was learned (**serious fault**). • The essay repeats information that is in the book or has been presented in class, without analysis or synthesis. • The essay correctly uses *vocabulary* • The learning described in the essay is primarily about code. Data is a second-class participant or is not mentioned at all.

	Exemplary	**Satisfactory**	**Must Improve**
Evidence	• The arguments made in the reflective essay are supported by the work samples, and everything in the work samples is relevant to the arguments made in the essay. • Where the two work samples are similar, the reflective essay points them out. • Evidence of learning includes at least one point of contrast in the two samples of work. (One point of contrast is sufficient; more contrasts are not better.) • The essay says what you built and how what you built shows what you learned. • The code in the samples works, and the essay mentions (briefly) how it is known that the code works. • Not all of the code in the samples works, but the essay explains what does not work and what was learned from the parts that do not work (or are not complete).	• Samples contain superfluous definitions, examples, or tests not needed to support the arguments presented in the essay. • The reflective essay points out one or more contrasts in the work samples, but it does not point out similarities. • The essay shows what you built and says what you learned, but leaves it to the reader to make connections. • The code in the samples works, but the essay does not mention how it is known that the code works. • Not all of the code in the samples works, but the essay is not able to draw any lessons from the fact of it not working.	• Samples lack the definitions, examples, and tests needed to support the essay. • The reflective essay does not identify points of contrast in the two work samples, *or* • The evaluators are unable to make the connection between the points of contrast in the work samples and the argument for learning made in the essay. • The portfolio shows something you built that really works, but not what you learned. • Nothing you built worked. • What you built isn't mentioned (**serious fault**). • The essay does not address the question of whether code works.