CS 161 Design and Peer Review Guidelines

Introduction

For each programming assignment (2 through 5), you will submit a Design Document explaining your proposed software implementation, then review two designs created by other students.

- The Design Document will be due one week prior to the programming assignment deadline. For example, if an assignment is due the Sunday of Week 4, then the design for the assignment must be posted on Canvas by 11:59 p.m. on the Sunday of Week 3.
- Your Peer Reviews will be due on Wednesday after the Design Document was submitted. For example, if the design document was due on Sunday of Week 3, then your Peer Reviews are due on Wednesday of Week 4 at 11:59 p.m. (also on Canvas). You will then receive reviews of your Design Document from other students.
- **NO DESIGNS ARE ACCEPTED LATE.** If you submit a design document late, you will receive a zero**. Additionally, you will receive a zero on the peer review grade.** For example, if you fail to turn in your design on time (by 11:59 p.m. on Sunday) to Canvas, then Canvas cannot assign designs to you for peer reviews, which means your penalty is ultimately a zero on the both design and peer review grades for that assignment.

Design Document (submit to Canvas)

Design is very important when developing programs, and there are a variety of ways to approach it. You may draw pictures, write it out in prose or structured text, use pseudo code, and more! The point of design is to give you a blueprint to follow while you are coding. This saves time later when debugging your program, since you can catch conceptual mistakes early (before any code is written). **It is better to spend one hour of time designing than it is to spend five hours debugging.**

George Polya developed a well-known model for problem solving in mathematics that is based on these 4 principles:

1. Understanding the problem. (*Recognizing what is asked*.)
2. Devising a plan. *(Responding to what is asked.)*
3. Carrying out the plan. *(Developing the result of the response.)*
4. Looking back. *(What does the result tell me? Did I do it right?)*

Polya's steps 1, 2, and 4 do not directly deal with writing the solution (in programming that is the C++ code itself), but rather, the steps you need to make sure you write a correct solution/program that solves the given problem statement.

1. Understanding the Problem

In your own words, explain what YOU think the problem is asking you to do. Document your uncertainties about the problem and anything else that you feel was unclear or

vague. Especially important: write down any assumptions you make (e.g., that a particular number must be positive or that the program should have 3 outputs

2. Devising a Plan/Design

Provide an algorithm/pseudo code to help solve the problem. In addition, draw pictures/flow charts to help you devise your plan, as well as any other design decisions you make, such as how to manage your time, how to decompose the problem, where to start first, etc.

4. Looking Back/Testing

Overall, this includes the process of testing as well as self-reflection that occurs while solving the problem. Examples: using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using sources of information to make sense of the results, etc. For the Design Document, you should identify Test Cases that show you have thought through how you want the program to behave. Each test case should include:

- Setting: What is the context? If the program is prompting the user, what is the prompt?
- Input: What happens in this test case? If it is a user prompt, what does the user type? If it is a file that is read in, what is in the file? And so on.
- Expected result: What do you want/expect the program to do when this test case happens?

For an example design, see this sample document (Links to an external site.):

Design Document Expectations

The design does not need to be completely correct, but it must show a good faith effort to create a quality design for the current assignment. It must address:

1. **Understanding the Problem (2 pts)**
    - What is the goal?
    - Do you understand everything in the problem? If not, what is not clear?
    - What assumptions are you making about the problem you are solving, user input, etc.?
2. **Program Design (4 pts):** What does the overall big picture of this program look like? Your choice:
    - Draw a **flowchart** (can be on paper/whiteboard – take a picture)
    - Write down the **steps** in pseudocode (must contain function names and headers)

    Do not submit actual C++ code. If you submit C++ code, then you have skipped the design step and gone straight to implementation.
3. **Program Testing (4 pts):** Test cases (must contain good, bad, and edge cases)

By default, you will receive one point for each area addressed in the design (up to 3 points for just turning in something). The remaining points for each area will be based

on how thorough and complete each section is. For example, consider section (1). Simply restating the problem will only get you one point. You must describe and justify your understanding of what the problem is asking you in order to receive 2 points.

Peer Reviews (access in Canvas)

In addition to submitting your proposed homework designs, you will also review and critique the designs of others. **Your peer reviews do not affect the other student's grade.** Instead, the goal is to provide constructive feedback that will help each student become a better programmer. (You will find that evaluating other designs will improve your ability to design programs, too!)

Peer reviews must be written **constructively and positively**. Negative, harsh, or mean comments are not appropriate and will cause you to lose points.

If your Design Document was submitted to the Canvas Assignment on time, Canvas will assign you two designs to review after the design due date on Sunday at 11:59 p.m. Look for "Assigned Peer Reviews" on the "Design Document" assignment page where you submitted. You must provide your peers with quality reviews focused on enhancing their design by 11:59 p.m. on the Wednesday after the design due date. For each review, you must **answer the specific questions included in the assignment instructions.**