Week 10 Tutorial

Modules

1. What is a module?
2. What do modules provide?
3. Describe how functions are modules.
4. What are the inputs and outputs to a module?
5. What is behavior that is private to a module?
6. What is the relationship between header (.h) files and modules?
7. Why do we design modules before programming?
8. What questions do we ask ourselves when refactoring modules?
9. What are the two things we list when designing a module?

ASCII Files

10. What is standard input?
11. What is standard output?
12. What is standard error?
13. How do you redirect standard input and output?
14. How do you create a file pointer?
15. What is the declaration for a file pointer?
16. What do the modes r, w and a mean?
17. What is the type of the argument to close()?
18. How do you read a character from a file pointer?
19. How do you read a string from a file pointer?
20. Write a function that copies the standard input to a named output file.
21. How do you check that fopen opened a file?
22. How do you check that fclose closed a file?
23. How do you write a character to a named file?

24. How do you write a string to a named file?
25. Write a function that copies a named input file to the standard output.

Binary Files

26. What is a binary file?
27. What do the following mean: rb, wb, ab, r+b?
28. How do binary files differ from ASCII files?
29. How are integers represented in an ASCII file? How are the represented in a binary file?
30. How do you open a binary file?
31. What is the function declaration of the function to write to a binary file?
32. Write a function that will add an integer to a binary file?
33. What is the function declaration of the function to read from a binary file?
34. Write a function that will read, then return an integer from a binary file?
35. How do you determine when you have read all of the records from a binary file?
36. What does the function `fseek` do?
37. What is the function declaration for fseek?