

Program Grading Guidelines

Your programs will be judged on several criteria, which are shown below, along with examples of questions that fall under each category. Note that while some assignments may be less demanding in some areas (for example, if you are given an ADT to use, there's not much question about design) the same point breakdown will be used for all programming assignments unless stated otherwise in the assignment.

Correctness (50 percent): Does the program compile correctly? Does the program do what it's supposed to do?

Design (20 percent): Are operations broken down into functions and procedures in a reasonable way? Are ADT's clearly defined, implementation independent, and sensible?

Style (10 percent): Is the program indented properly? Do variables have meaningful names?

Robustness (10 percent): Does the program handle erroneous or unexpected input/situations gracefully?

Documentation (10 percent): Do all program files begin with a comment that identifies the author, the contents, and the compiler used for that particular file? Are all the procedures and functions clearly documented using Javadoc-style comments? Are unclear parts of code documented?

Notice that correctness is only half of the grade. Design and style are very important: you write an English paper by first laying out how you want the information presented, then writing a rough draft that makes all your points, and then polishing the style of that rough draft. Writing a program is much the same: first design how things should be organized, then do a "rough draft" that consists of a working implementation, and then go back and polish your code, adding all the required comments and make sure style guidelines are followed. Unfortunately, many people hand in their "rough draft" programs, thinking that a working program is all they need. If you do that in this class, you probably will not pass.

A program that does not compile will get at most 50% of the possible points for an assignment. If you add comments after getting your program working, then be sure you test it again before submitting it. It is actually fairly easy to make working code inoperable when adding comments (by forgetting to close a comment, for example). Even if your program is entirely correct except for a small mistake of this type, you will still get a maximum of 50%.