16.216: ECE Application Programming

Program Grading Guidelines

1. Acknowledgements

A significant portion of this document was copied directly from the UMass Dartmouth ECE 160 rubric (as of September 2011), with some minor modifications. Thanks to Professor Phil Viall and Ben Viall for allowing me to use their work.

2. Grading Rubric

This rubric will be used unless explicitly specified otherwise in the program handout. Although some of these items may seem "picky," they are included to ensure that programs can be graded in a reasonable amount of time.

Please note that many points described under "Appropriate documentation" and "Appropriate looking code" are discussed in detail during lecture and eventually will be listed in the 16.216 Programming Style Guide, which is still under construction.

Correct submission name..**10%**
Each program has a specified name that must be used when submitting your files. This name will be listed in the assignment, typically in the form "`prog#_progname.c`" (e.g., `prog1_simple.c`). If a program requires multiple files (for example, header and source files), multiple filenames will be listed. The name(s) you use must match the given name(s) exactly. The quote marks are not part of the name.

Appropriate documentation..**10%**
Appropriately written code is often self-documenting, but your files should meet the minimum commenting requirements described in the 16.216 Programming Style Guide: namely, a header comment for the entire file as well as a comment for each variable declaration, related block of code, and function definition.

Appropriate looking code...**20%**
This section covers issues related both to program content and style. Appropriate looking code is code that appears (to me) related to the given problem. In addition, your code should be stylistically appropriate as well, in terms of indentation, variable/constant naming, and other points discussed in the 16.216 Programming Style Guide.

Successful compilation **10%**
The code compiles without any fatal errors/warnings that prevent it from working.

Correct output **50%**
Output is as specified in the program specification, including both correct values and formatting. Do not simply try the sample inputs to see if your program produces the specified output. I expect you to try additional possibilities; I assure you I will try several data sets which are not in the handout. Your program should function as specified in the problem statement. Incorrect output results in lowering of score.

3. Common Deductions

Submission after due date ....-(2^{n-1}) points
where n is the number of days late. In other words, a program that is one day late loses 1 point; a program that is 4 days late loses 8 points. A program that is more than seven days late will receive 0 points. Programs are due on the date specified by 11:59:59 PM. A "day" is defined as a 24 hour period beginning at midnight, and includes Saturdays, Sundays, and holidays.

NOTE: *During summer terms, the late penalty is -(4^{n-1}) points, to account for the accelerated course schedule.*

Program does not compile...-60%
While the above rubric states that successful compilation is worth 10% of your grade, programs that do not compile cannot be tested for correct output, which is worth 50% of your grade. Code that does not compile is therefore penalized for both types of errors.

It may seem unusually harsh to deduct 60% for a submission that may have only a single compilation error. However, I think it's fair to deduct points for incorrect output when I know you haven't tested your own program to see if it runs correctly!

Incorrect input structure ……... up to -20%
In most projects, the input interface (commands, data format/order, etc) is part of the specification, and must be followed to avoid loss of credit. Making the program more "user friendly" will not result in a higher score—it will likely lower your score.

Incorrect name for associated data file …….. -10%
Some programs will require your program to read and/or write a file. The name of the file will be specified in the handout. Your program must use this exact file name.

4. Regrade Policy

For each assignment, you will be allowed one resubmission to improve your grade without penalty. Each additional resubmission carries a one day late penalty. Note that:

- Each penalty-free resubmission has a deadline based on when the original assignment submissions are graded. Late penalties apply to late resubmissions.

- The resubmission policy does not allow you to avoid penalties when the original submission is late. For example, an assignment losing 4 points for a late initial submission has a maximum possible score of 96 for the resubmission.

- In order to take advantage of the penalty-free regrade, your original submission must demonstrate a significant, real effort to complete the assignment. A file with little or no code related to the assignment does not count as a submission.

  In other words, do not turn in a virtually empty file simply to avoid late penalties. The late penalties are based on the date on which you first submit a program containing a substantial amount of relevant code.

5. Grading Examples

This list contains examples of grades you will receive if your program fits the given description. The list is not exhaustive and assumes a maximum score of 100 points.

- You hand in a perfect program *(during the fall or spring, not the summer)* ...
  - 1 day late..........99 points [-(2^{1-1}) = -1]
  - 2 days late........98 points [-(2^{2-1}) = -2]
  - 3 days late........96 points [-(2^{3-1}) = -4]
  - 4 days late........92 points [-(2^{4-1}) = -8]
  - 5 days late........84 points [-(2^{5-1}) = -16]
  - 6 days late........68 points [-(2^{6-1}) = -32]
  - 7 days late........36 points [-(2^{7-1}) = -64]
  - 8+ days late........0 points [-(2^{8-1}) = -128]
- You hand in a perfect program with the wrong name..............................90 points
- You hand in a perfectly running program with no documentation.............90 points
- You hand in a program that is well documented, compiles with no errors, and runs, but it produces no output ....50 points
- You hand in a correctly named program with relevant code and acceptable documentation that does not compile..40 points
- You hand in a largely correct program 3 days late, and your output is approximately 80% correct.......................................86 points (-4 late, -10 output)