

# Grading Policies

## CMSC 471 – Fall 2011 – Prof. desJardins

*All students must read, understand, and follow the course policy on academic honesty and grading standards. Each student will be asked to sign a statement indicating that they have read and understood the policy.*

### 1 Policy on Solution Keys

Solution keys are for your personal use only, and are to be destroyed after you review them. In particular, you may not give solution keys to students in other classes, or who take this class in future years.

### 2 Late Policy

Each assignment will have a due date and is expected to be turned in on time. Extensions of up to one week may be granted on an individual basis by the instructor, if requested in advance. Repeated requests for extensions, or requests for extensions at the last minute, will be denied other than in extraordinary circumstances.

Homeworks will be at the beginning of class on the due date. (That is, an assignment due on a Thursday will be due at 2:30pm on Thursday, and must be turned in in class. If you will miss class, you must arrange for your homework to be turned in to the instructor or TA during or before class.) Each student will have one free late—a one-week automatic extension that can be used for any single homework over the course of the semester. Please let Dr. desJardins know you are using this extension on the day that the assignment is due, and write “EXTENSION” on the assignment when you submit it. Requests for additional extensions will be denied other than in the most extraordinary circumstances.

### 3 Grading of Programming Assignments

This is an *approximate* distribution of how your grade will be allocated on programming assignments. The specific percentages may vary some from one assignment to another.

- A *correct* solution (i.e., one that returns the right answer in all cases) will receive 80% credit. Note that you must have reasonable error checking to receive this credit.
- A *readable* solution (i.e., one that is commented with a header comment, documentation line, and/or inline comments as appropriate, *and* properly indented) will receive another 10%.
- An *elegant* solution (i.e., one that is simple, clean, efficient, and understandable) will receive another 10%.

**Regarding Correctness** No matter what arguments are passed in, your solution should not break. You may either return an error code, return a default value, or use `error` or `cerror` to signal an error and enter a break level. In any case, the error action should be clearly documented.

**Regarding Readability** In solution keys, I will typically provide header comments, a documentation line, some inline comments, and properly indented code. This is perhaps more commenting than a simple program needs, but it's a good habit to get into anyway.

When you start to write longer programs, with many functions (especially if the functions are small), rather than a detailed header comment for each function, you may want to group the functions together by category, including a header comment for each category.

Please think carefully about how to organize and document your code so that it is readable. Developing software is about more than just getting the right answer: it's about writing code that is understandable, extensible, maintainable, and modifiable by yourself *and* by others who may use your code after you.

**Regarding Elegance** Solutions that are inefficient, take many more lines of code than necessary, or use lots of temporary variables when just a few would do, are not as easy to understand or maintain as *elegant* solutions. On the other hand, elegance doesn't just mean "short functions," since making functions very short may also make them obscure and difficult to understand. Elegance in programming is an art; the only way you can learn it is to think carefully about how you formulate your solutions, and to study examples of well designed code.

**Regarding Submissions** We reserve the right to run your homework assignments on previously unseen test cases, so you should be sure that (a) you use the function names that are given in the homework, and (b) you ensure that your code works on unseen cases, not just the test cases we ask for in the script.

## 4 Grading of Written Assignments

Some of the written assignments will have a "right" answer. Those will be graded according to correctness, with partial credit given for incorrect solutions to the extent that you've shown your work and indicated why you believe your answer to be correct. All answers should include a clear justification: a correct final answer with no explanation of how the answer was obtained will only receive partial credit.

Other assignments will require you to express opinions in short answers or in essays. As with programming assignments, a portion of your grade will be given for the content of the essay, and a portion will be given for readability and style. As with programming assignments, the *approximate* distribution of credit will be as follows:

- 80% for content (well thought out and well reasoned answers; answers that are "correct" to the extent that there *is* a correct answer, which there often is not).
- 10% for "readability": correct grammar and spelling, readable formatting.
- 10% for "elegance": well expressed thoughts in a well structured essay.

All written assignments must be typed or very legibly handwritten, and must be proofread with reasonable attention to spelling, clarity, and grammar. It is disrespectful to the course staff to submit an illegible or poorly prepared assignment. Illegible assignments and assignments with large numbers of typographical and grammatical errors will be returned without a grade; to receive a grade, the assignment must be resubmitted in legible form by the next class period. Only one such resubmission will be permitted per student per semester.

Although this is not a writing class, success in any scientific discipline requires the ability to effectively communicate one's thoughts. If you have difficulty writing, whether it's because English is not your first language, or because you haven't taken many writing classes in your undergraduate program, I highly suggest that you take advantage of UMBC's writing center, in the main library. (Phone: 410-455-3126. URL: <http://www.umbc.edu/lrc/writing-center.htm>. Hours: Monday – Wednesday, 10 a.m. – 7 p.m.; Thursday, 10 a.m. – 5 p.m.; Friday, 10 a.m. – 2 p.m.) This is a free tutoring service that will help you prepare essays and papers for any course.

## 5 Academic Honesty

This course adheres to the Provost's statement on academic integrity:

“By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the UMBC Student Handbook, the Faculty Handbook, or the UMBC Policies section of the UMBC Directory.”

Cheating in any form will not be tolerated. In particular, examinations are to be your own work. You may discuss the homework assignments with anyone. However, any help you receive must be documented. At the beginning of your assignment or program, you must explicitly indicate the sources you used while working on it (excluding course staff and text), and the type of help you received from them. If you do not include such a statement, the course staff will assume you worked entirely independently. Any indication of collaboration with other students in this case will be considered a violation of the academic honesty policy.

The implementation of the programming assignments must be your own work. If you are stumped on a particular error, you may consult with someone else; however, if you consult with someone other than the instructor, the TA, or the help desk, you must place a comment in your code near the point of the error, stating the source and scope of the help you received. Reasonable help will not affect your grade; failure to cite your sources is academically dishonest, and will be dealt with harshly.

Written answers on essay questions for homeworks and exams must be your own work. It is entirely acceptable to discuss the homework assignments with other students, but the actual answers should be *your own* answers, not group answers that are copied down. If you wish to quote a source, you must do so explicitly at the point of the quotation, with proper citation. Plagiarism of any source, including another student's work, is not acceptable. If you wish to quote a source, you must do so explicitly, **using quotation marks and proper citation at the point of the quote**. A useful guideline is that if more than two or three words in a row are the same in your report as in the original source, you have plagiarized.

<http://owl.english.purdue.edu/owl/resource/560/02/> gives an excellent overview of the APA style for correctly citing a source. (You may use any style as long as your bibliographic entries are complete and correct.)

<http://www.indiana.edu/wts/pamphlets/plagiarism.shtml> gives guidelines on acceptable paraphrasing. (See also the other documents at <http://www.indiana.edu/wts/pamphlets.shtml> for useful suggestions on writing and citing sources.)

**The same requirements apply to programming assignments.** Your programs are expected to be your own, original work, and not borrowed, copied, templated, or modified from code that you have obtained elsewhere. I recognize that code reuse is a legitimate form of software engineering—assuming that licensing and copyright issues are addressed—but it is not appropriate for assigned coursework. Under no circumstances should you show your program code to another student.

**Fabrication:** Fabricating results, sources, or any information, is academically dishonest and subject to the penalties outlined below.

**Aiding and abetting:** Providing another student with answers, or helping them to cheat, is an **equally serious** violation of the principles of academic honesty. A student who commits such an offense is subject to the same penalties as the student who cheated.

**Penalties:** Any violation of the academic honesty policy will result in a minimum penalty of a zero grade for that assignment. In addition, in order to pass the course, the student will be required to recomplete the assignment honestly. Consequences for more serious infractions of

this policy, or for second offenses, may include, but are not limited to, receiving a failing grade in the course or being suspended or expelled from the university.

---

---

I have read and understood the CMSC 471 course policy on academic honesty and grading standards. I agree to follow the policy, and I understand the consequences for violating the policy.

---

Signature

---

Date

---

Print Name