Academic Honesty Policy and Grading Standards

CMSC 671 – Fall 2005 – Prof. desJardins

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.

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 Academic Honesty

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. [Statement adopted by UMBC's Undergraduate Council and Provost's Office.]

Cheating in any form will not be tolerated. In particular, all assignments are to be your own work. You may discuss the concepts, ideas, and what is being asked for in the assignments with anyone. However, when you sit down to write your answers or code your program, you must write it yourself, using your own ideas. If you simply write down an answer that somebody else gave you, it is dishonest and will be considered a violation of the policy.

Written answers for homeworks and reports must be your own work. Plagiarism (copying) 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**.

`http://www.lib.duke.edu/libguide/bib_journals.htm` gives an excellent overview of how to correctly cite a source.

`http://www.indiana.edu/~wts/wts/plagiarism.html` gives guidelines on acceptable paraphrasing.

The implementation of the programming assignments must be your own work. You will be using a public-domain software package as a starting point, but your application must be your own work. Before using code or knowledge bases from any source, you should check with the instructor to be sure the use is acceptable. If you do use code from another source, you must explicitly cite the source. If you are stumped on a particular error, you may consult with someone else; however, you must include an acknowledgement of the source and scope of the help you received in your project report. Reasonable help will not affect your grade; failure to cite your sources is academically dishonest.

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: The consequence for any infraction of this policy is, **at a minimum**, a zero grade **for the entire assignment**. In addition, in order to pass the course, the student may 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.

2 Grading of Assignments

Late Policy: All students are responsible for reading and understanding the late policy on the course website.

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 complete 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%.

3.1 Regarding Correctness

No matter what arguments you pass 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.

3.2 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.

3.3 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.

3.4 Regarding Submissions

I 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 I 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.

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: 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.