

# Elmer W. Engstrom Department of Engineering and Computer Science

## Computer Science Graded Work Policy

Version 1.8: September 7, 2016

**Note:** Where the policies in this document differ from those found in course syllabi, the policies found in course syllabi take precedence.

The purpose of this document is to explain the degree to which you may get help on graded work (e.g., homework, quizzes, reports, programming assignments, etc.). The comments below are not intended to be exhaustive, i.e., all that can be said regarding this subject—they are intended to be helpful guidelines for the conscientious student who wants to maintain his/her integrity. Scripture commands us to “abstain from all appearances of evil” (1 Thes 5:22) and to “provide things honest in the sight of all men” (Rom 12:17b). Also, note that integrity failures violate the University Academic Integrity Policy. As a general principle, it is best to always be squarely honest than to test the “edge” of right and wrong. In order to help ensure your integrity, we freely invite you to ask questions regarding these policies on any issue that may be unclear to you.

It is plagiarism to copy ideas, text or code from any source, including the internet, without proper citation.

In this policy, we distinguish between three types of effort in graded work: **individual effort**, **collaborative effort**, and **group effort**. These three types are explained below.

**Individual Effort:** Individual effort means that a student must complete the graded assignment entirely on his/her own without help from any other student (current or former at this University or another), faculty member, or parent with the exception of current Cedarville computer science professors. Individual effort precludes students from using published or on-line resources, unless specifically allowed by the faculty member giving the assignment. Where these resources are allowed, uses of such work must be cited. **Individual effort is the default policy for assignments when no specific written guidance has been provided by the course instructor. In the absence of any specific instructor guidance for an assignment, you should assume the graded work policy for the assignment is individual effort.**

**Group Effort:** Group effort means that students are fully permitted and expected to produce a joint answer to a given problem. Please recognize the size of the “group” on a group effort problem may be limited. Between groups the default policy is individual effort. **The group effort policy only applies when explicitly given in connection with a specific assignment by an instructor in written form.** When completing group and team projects it is expected that each member of the group will contribute significant effort. It is also expected that each member of the team understands any portions completed by other members of the team. We reserve the right to assign different grades for each member of the team when the effort or learning is very different among the team members.

**Collaborative Effort:** Collaborative effort means that a student must complete the graded assignment *principally or substantially* on his/her own. Recognize that along the spectrum of effort from individual to group work, collaborative effort should be viewed as more closely associated with individual effort than group work. **Collaborative effort is only allowed on a specific assignment, if the instructor has provided written guidance clearing stating that this is the graded work policy for the assignment.**

Generally, we encourage students to discuss the overall intent and purpose of collaborative effort assignments for the purpose of education and understanding about the subject to which the assignment pertains. In addition, it is valuable for all students when stronger students assist weaker students in getting past problem misunderstandings, compilation errors, or difficult syntax. The scriptural basis is in Rom 15:1 and Gal 6:2 where the strong are encouraged to assist the weak.

However, when we give or take *too* much help, the weaker students are not ultimately helped, but hurt, because they are not held accountable for producing their own work. So we have the principle that we should each, in part, bear our own burden (Gal 6:5).

The intent of this policy is to help maintain a right balance among those that give (the “helper”) and receive (the “helped”) help. As faculty, we allow a helper to assist a helped so long as they can do so without revealing a substantial portion of the solution. To uphold this intent, please follow this guideline. **Under no circumstance**

**should the helper give or assist the helpee in obtaining a complete (or substantially complete) solution in any form: written, electronic, or verbal.**

The bottom line is that collaborative work is substantially *your own* work. Not another's. Not group work.

**Programming Assignments:** With regard to programming assignments, we understand that it is sometimes difficult to draw the line between educationally useful sharing of ideas and educationally destructive copying of ideas. Therefore, this section is intended to give additional guidance for these types of assignments, beyond the three types of effort covered above. Our goal is to maximize the educational benefit of programming assignments. We do not want students to get stuck and waste a significant amount of time on a simple syntax issue (e.g., where to place a semicolon); however, we want to ensure that students do not receive so much help on an assignment so that they don't learn the material it is trying to teach. The paragraphs below provide general guidance on how much information sharing is acceptable. If you need more specific guidance as to whether something would be permissible, please see your instructor.

- A. **Collaboration that is always allowed:** Listed below are actions which we encourage on programming assignments.
- Discussing material covered in lecture, section or the text.
  - Discussing the requirements of the assignment.
  - Discussing features of the language (C++ or Java) and its standard libraries.
  - Asking specific questions about syntax or specifications. For example: "Is strcmp case-sensitive?" or "What does the keyword static do here?"
  - Discussing general techniques of designing, coding, or debugging. For example, saying things like "It worked well to test each function right after I wrote it." or "When my programs crashes, I first look at the stack trace in the debugger." is fine.
  - Discussing features of any of the programming tools or development environments.
  - Any discussion between the student and the instructor. You are welcome to discuss any and all ideas, design, code, debugging, and details with the instructor, and to get advice from the student aides in the lab. In general, the lab aides are better to talk to than classmates, because they are knowledgeable about all the material and know how to help you without giving away too much.
- B. **Collaboration that is NEVER allowed:** Basically, the rule is that you should be handing in code which represents your original, independent work. It should not be based on, influenced by, or copied from the work of anyone else, including classmates or students from prior semesters.
- **Copying code.** This is the most blatant violation. You should not be writing down anyone else's code, or allowing anyone else to write down your code.
  - **Using work from past semesters.** Using someone's work or solutions from a previous semester is an obvious violation. Additionally, students must receive permission from their instructor to directly use work they developed in a previous course.
  - **Looking at someone else's code.** You should never read anyone else's code whether it is on the screen or written out by hand, except as discussed below in the **Debugging Guidelines**.
  - **Stealing someone else's design.** Discussing design with someone else and sharing ideas and critiquing each other's design is okay, if attributed. However, just taking someone else's design without trying to develop the design yourself is not allowed. It is akin to taking someone else's outline for a research paper and basing your paper on that.
  - **Asking for help on something you haven't thought about yourself.** Always make every attempt to tackle a problem yourself before asking another student or teacher for help. It will help you to become a better and more self-sufficient programmer, as well as a better student. The best way to learn is to try!
- C. **Collaboration that is allowed, if documented:** Whereas high-level abstract discussions are always allowed, two students engaging in a more detailed discussion of a particular design decision, algorithm, or a student helping another to track down a bug in a manner that requires the helper to read the helpee's code will cross into the area of collaboration that is acceptable only if documented. We require that you include the name of any student(s) from whom you received such assistance and properly credit their contribution to your work. This is akin to acknowledging a reference in a research paper. Some examples:
- **Discussing the design or general solution of an assignment.** Design is a crucial part of the programming process, and discussing it is very valuable. However, learning to work through design

problems on your own is a skill which requires time and practice. Try to work out as much as you can on your own, but discussion amongst students is okay, as long as it is documented.

- **Helping another student to debug a particular problem.** Two students should not sit down and debug jointly, but one might give the other some direct hints (e.g. “segmentation fault usually means you are accessing your array outside it’s limit” or “when I had a similar problem, it was because I had forgotten to put a terminating null on my linked list”). In general, we prefer that you get this sort of help from the lab aides (who know better how to guide you without going overboard), but if you do get detailed debugging advice from someone, you should credit their assistance. See **Debugging Guidelines** below for further guidance.
- **Sharing advice about testing.** For example, if your roommate tells you about some lesson learned (“my program didn’t handle the case where the input file didn’t end with a newline”) that you then use to improve your program’s robustness, you should credit your roommate for providing you with that insight.

**Debugging Guidelines:** The goal of these debugging guidelines is to provide some mechanism for students to become “un-stuck” without students receiving so much help that their learning is inhibited. However, we also understand that once you start giving someone assistance with their program, it can be easy to go too far, so please carefully follow the guidance below.

Assuming you document the help, it is permissible to assist someone in fixing a difficult bug. Every effort should be made to ensure the person being helped understands the problem and would be able to solve a similar problem for themselves in the future. You should limit help to assisting with one or two bugs at a time; you should not sit down with someone and debug their entire program with them. Moreover, you should only help them after they have made a real effort to solve the particular problem on their own.

Assistance in debugging involves the implicit roles of the “helper” and the “helpee,” and these roles should not be exchanged on a single assignment. The helper may look at portions of the helpee’s code for the purpose of assisting with debugging, but not with the purpose of learning something that the helper can use in his/her own program. The helpee should **never** look at the helper’s code. Looking at another student’s code should be done with care, and only for the purpose of helping them past a difficult spot. It is acceptable for the helper to tell the helpee what is wrong, but they should refrain from offering a correct coding sequence; that is, the helper should allow the helpee to determine how to correct the error on their own.

Common violations of the above guidelines are:

- when the helper shows his/her code to the helpee,
- when the helper coaches the helpee line-by-line to a solution,
- when the helper does not make reasonable effort to guard access by the helpee to her/her own work (e.g., provides access to a written or emailed solution).

**Internet Usage:** The Internet has become ubiquitous in our society and provides enormous benefits for communicating and sharing knowledge. Information is literally available at our fingertips. However, we all know that misuse of the Internet poses dangers to us as Christians. Like many things in life, it can be both a blessing and a curse depending upon its usage. In academics, the use of the Internet can serve to increase or decrease learning. As a source of information to supplement our textbook or lectures, it can help students to grasp difficult concepts. However, using information from the Internet to replace careful thinking through a problem results in decreased learning. Use of the Internet also raises the danger of plagiarism, unless we are very careful to document our sources.

Clearly, the Internet should never be used on exams or quizzes. For other assignments such as homework and programming projects, the Internet should never interfere with the student’s responsibility to develop a solution substantially on their own. Students must carefully evaluate their use of the Internet and ensure that it will impact neither their integrity nor their learning. Wisdom dictates that students err on the side of caution in their use of the Internet. Some examples:

- **ALLOWED** – Looking up how to seed a random number generator in C++, when this is not the principle purpose of the assignment.
- **ALLOWED** – Looking up how to use the HashMap class in the Java libraries.

- **ALLOWED** – Looking up the general concept of a Linked List so that you can develop code to implement one.
- **NOT ALLOWED** – Looking at pseudocode or actual code which performs an insert into a Linked List, when your project requires this to be implemented as a substantial portion of the assignment.
- **NOT ALLOWED** – Searching for a specific problem number from your textbook which is assigned as a homework problem.

**Summary:** It is essential that you always make choices which do not compromise your integrity or even give the appearance of doing so. If you are going to receive help, please consider: guard your integrity carefully---we generally know when we have passed from learning to simply copying what is shown, given or recited to us. Please exercise responsibility for your learning. While copying solves an immediate need, it is dishonest and lays a poor foundation for the future success.

**If you find you need help to complete the majority or all of the assignment, then you are in need of additional instruction and need to make an appointment with your instructor.**