

Honor Code for Assignments in Introductory Programming Courses at Bilkent CS Department

In the Computer Engineering Department at Bilkent, we take academic integrity on assignments very seriously and expect you to take it seriously as well. The aim of this document is to make our expectations as clear as possible to prevent possible academic integrity violations.

Students naturally want to work together and get help from others when they struggle. We believe that, as a student, you learn a great deal by doing so and encourage such assistance. However, **performing certain tasks on your own is a crucial requirement for learning certain aspects of programming** [1]. For instance, discovering a syntax error in your code might take a long time for you as an inexperienced programmer. An experienced coder or an AI-based tool pointing this out to you does solve the problem much quicker, and you might understand your mistake. However, the amount of learning that takes place isn't the same in these two scenarios (no pain, no gain!) [2] Similarly, one can easily learn how to calculate the median of a set of unordered numbers efficiently if s/he is given the algorithm for it. However, trying to actually come up with an algorithm of your own is a much longer process, teaching you much more about developing similar algorithms in the long run. Because of this very reason, you should be careful not to violate academic integrity. Below you will first find a summary of rules, then detailed explanations with examples.

Rules Summary: Acceptable (✓) vs. Unacceptable (x)

Individual Assignments

✓ Reuse code given in textbook and/or lecture material (slides, notes, etc.)	X Reuse code from the Internet, previous years' solutions, friends, etc. For example, getting the code for a linked list from the Internet and reusing it in your assignment is not acceptable.
✓ Reuse small code segments (at most several lines of code) with proper attribution/citation for minor calculations such as the calculation of the intersection of two line segments.	X Make use of sources other than course material (even if it's allowed) without providing a reference to the source. X You may be assigned the same problem from a previous year, but in case that happens, you should not make reuse of any available solutions.
✓ Discuss strategy, design, architecture, and algorithm (not at code or pseudocode level!) with others	X Discuss or debug your code or pseudocode line by line with others (except the course instructor or TA).
✓ Ask a classmate or a more senior student about the requirements or strategies that might be used in coding a programming assignment (not at the code or pseudocode level)	X Get help from classmates or other individuals such as a more senior student or automated tools (at code or pseudocode level).
✓ Get help from the course instructor or TA with your code.	
✓ Hire a tutor to teach you concepts (not do your assignment for you or help you do it).	X Hire a tutor to do your assignment or help you do your assignment.
✓ Explain to a friend the requirements, design, or architecture suitable for a programming assignment.	X Helping someone else in his/her programming assignment (at code or pseudocode level) even if you are not getting any help from him/her.

✓ Direct a struggling friend to your instructor or TA.	X Help a struggling friend “a little”, not much!
✓ When unsure about what is acceptable, ask your instructor.	X When unsure about what is acceptable, ask somebody else (other than your instructor).
✓ Keep your solutions to assignments in a private repository	X Make your solutions to assignments or other material from previous years available in a public location (e.g. a GitHub repository)

Group Assignments

✓ When working within a group, I can take more responsibilities in certain types of tasks (that I am good at) and do less in other types of tasks (e.g. do more coding than others and do less documentation).	X When working within a group, I can perform only certain types of tasks since I am more talented / experienced in doing those and I don't have to do any other type of tasks (e.g. do only coding, no documentation or analysis, no design activities).
✓ When working within a group, even though I am an outstanding student with excellent technical (hard) skills, I also have to be a good team player, listening to others, valuing their opinions, and cooperating with them.	X Since I am an outstanding student, I get to make all decisions in group assignments.
✓ I have to honestly report activities I or other members of the team performed (to the best of my knowledge) as part of my group project.	X I don't have to tell my instructor what activities I performed as part of the group project.
✓ I cannot pretend that I actually performed tasks towards the group assignment even though I didn't. If I do so, then I might be held responsible for any violations of these rules. All team members are responsible for the whole work of the group even if they did not participate in some of the tasks involved.	X I am not responsible for any violations of these rules in a group setting since I didn't participate much in the group project (even though I didn't report this fact to the course instructor).

Rules

Below you will find detailed explanations of rules for conducting individual and group programming assignments.

Individual Assignments

Unless otherwise stated, assignments are for individual students with the following rules:

- You are allowed to reuse the codes given in your textbook and/or your lecture slides. However, you are **not** allowed to use any code from somewhere else (e.g., from the Internet, other textbooks, or other slides).
- Unless otherwise stated, you are allowed to discuss with your classmates in finding strategies, designs, or data structures for problem-solving and debugging, but **not** in writing or debugging your codes, including pseudo-codes. In other words, you may not discuss with your classmates how to solve the problem at the code level (using a particular programming language) or at the pseudo-code level. **Discussion or sharing at the actual code or pseudocode level will not be tolerated.**
- All work on assignments must be done individually unless stated otherwise (i.e., group projects). You may not copy code in whole or in part from someone else. You may not share your code. You may not discuss your solution with others in detail (line-by-line, loop-by-loop, etc.). If you need detailed help, ask your instructor or your TA. Such detailed discussion is limited to only these people. Other than these people, you may not receive detailed help on your code from individuals or books (other than the textbook) or the Internet unless stated otherwise.

- Similarly, you may not hire a tutor to get your homework done. If you just use some “small hints” from other sources (friends, books, websites, etc.) other than those prohibited here (e.g. how to calculate the square root of a number or to calculate the intersection of two lines), do not forget to cite them.
- If external assistance (from other people, the Internet, or books) is allowed for a particular assignment, you must indicate the source of your submission. If you make use of such assistance without giving a proper citation, this will constitute plagiarism. It is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary of having someone else write or debug the actual code as explained earlier.
- You are not allowed to look at solution sets or program code from other years. It is a violation of academic integrity and it will simply save the day but will not help you in the long run. Whenever you seek help on an assignment, your goal should be improving your level of understanding and not simply having your homework done.
- You may not show your code to others as a means of helping them. Sometimes, good students provide their code (sometimes “just a peek” at their code) or detailed help to struggling students with the intention to help them. However, such good intentions are violations of academic integrity. If you have such struggling friends, direct them to the instructor or the TA.
- You may not leave your code in publicly accessible areas. You may not share computers (at least the same account on a shared computer) in any way. If you work in a public lab, delete all files related to the assignment from publicly available drives before you leave, and make sure to log out. Do not forget to empty the recycle bin. You may not use others’ storage devices (others’ CDs, USBs, etc.) to save your work.
- You may not share your code publicly in online repositories (e.g. GitHub) and may not have online public manuals/tutorials of any kind which explain your solution. However, having a private online repository that organizes your projects is a good software engineering practice.
- You may not leave any printouts lying around somewhere and you may not dispose of them in public trash cans. It is your responsibility to keep your work from others to avoid disciplinary action.
- You must be prepared to explain the solutions to any assignment you submit. All the comments regarding the program code above are also valid for any essays you write and any theoretical solutions you are asked to provide.
- You may not rely on the assumptions regarding this issue (the experience of your friends or your experience in other courses). When you’re in doubt about what is acceptable and what is not, simply ask your instructor.
- The course instructor or TAs may use an automatic code comparison tool to help spot assignments that have been submitted in violation of these rules. Such tools take all assignments from all students (and solutions that may exist on the Internet or solutions from past years) and compare them. Assignments that are likely to be too similar are then carefully investigated for cheating or plagiarism. Notice that such tools can detect similarities in code even when variables are renamed, indentation and commenting are completely different, or code segments, functions, etc. have been moved around.

Group Assignments

When you’ve been asked to work in a team for an assignment/project, those rules that apply to individuals as described above will now apply to the team. In other words, when working in a team, you must obviously talk with, discuss, and cooperate on your assignment with your teammates but not other individuals or teams. **Additionally**, to get a passing grade and avoid any disciplinary action, you must obey the following:

- If the assignment requires multiple types of tasks (analysis vs. design vs. coding vs. documentation), then an individual must participate in and meet minimal requirements for each such task. In other words, an individual

who likes coding and hates documentation (or one who is not into coding but can write documents well) might fail the assignment if they don't participate in **each phase** of the assignment above at least partially.

- In real life, professionals are expected to be good team players. A student who is technically very good (hard skills) but fails to be a team player (soft skills) is likely to fail team projects.
- When inquired, a student is responsible for reporting tasks/activities he/she or others (to the best of your knowledge) performed to the course instructor or TA in an honest manner.
- If you do not participate enough in the group project to be aware of any violations of these rules but pretend that you fully participated in the project, you might be held responsible for any violations of these rules (at least for being dishonest with your instructor about the grading of the project).

Students caught cheating on assignments will be subject to disciplinary action without any exceptions. Do not forget that it is much easier to explain a poor grade to others than to explain a cheating conviction.

[1] [Learning rewires the brain](#)

[2] [The Impact of AI on Computer Science Education](#)