

GUIDELINES FOR PROGRAMMING SESSIONS FOR CS 213M

1. The practice sessions are meant to reinforce the theory and analysis aspects of the course material that is covered during the lectures.
2. Program development is an essential component of this course. Much like many other skills in real life, cycling, swimming, etc., no amount of theory will help in performing these acts smoothly in practice.
3. The practice part of the course has been introduced in this offering to give students an opportunity to sharpen their programming skills in C++, over and above those acquired in CS 101 or an equivalent course. The real take away of CS 213 is critical awareness of various design alternatives available for writing algorithms for a given application. The Teaching team has decided to take extra effort in the formulation and evaluation of programming problems to respond to this requirement.
4. IT IS IMPORTANT TO NOTE THAT CRUDE BUT WORKING PROGRAMS ARE ACCEPTABLE AS COMPARED TO PROFESSIONAL ONES THAT ARE COPIED / BORROWED FROM OTHER SOURCES.
5. YOU AND YOUR GROUP ARE URGED TO DESIGN AND IMPLEMENT YOUR PROGRAMS COMPLETELY BY YOURSELF. COPYING FROM EACH OTHER OR OTHER SOURCES WILL RESULT IN AWARD OF **FR** GRADE AND IMMEDIATE EXIT FROM THE COURSE AS SOON AS IT IS DETECTED.
6. Wait for instructions from your TA for details about method of uploading your program and other documents to be submitted, the DEADLINES, and such information.
7. It is desirable that your programs abide by the following norms :
 - Good indentation and brief comments at important points in the code for ease of readability.
 - A consistent scheme for naming all entities – class, object, variable, functions, etc.
 - Function body should be of moderate size and a definite purpose. The arguments and return type are to be chosen carefully.
 - Use interesting features of C++ all through : file i/o; function/operator overloading; template functions and classes; etc.
 - Feel free to define as many classes; non-member functions; as you deem appropriate.
 - You may be constrained to use only certain features of data structures / C++ in a given assignment. Read the instructions and honor the constraints. The general principle is not to use features that has not been covered till that point in time in the course, unless mentioned otherwise.
 - In case the program size is large, say > 500 lines, good practice is to distribute the code in multiple files, each of moderate size and cohesive content.
 - The main() function should be brief and essentially uses the classes and methods to solve the problem.
 - Use Linux and g++ as your implementation platform. Make sure that your program is free from all errors, compilation and run time, and test out the executable with inputs of small size before submission.
 - In case you wish to run the executable with various inputs for performance measurement, the input may even be generated by a program, a shell program to control the sequence is advisable.

HAPPY PROGRAMMING

S Biswas