CPSC 1375 Programming I Spring 2017 Course Outline

Sect 01: TuTh 1:40 pm - 2:55 pm, EIT 325.
Sect 11: TuTh 6:00 pm - 7:15 pm, EIT 325.

Instructor: Dr. S. Minsker, office EIT 575, office phone 569-8131, email sxminsker@ualr.edu, office hours TuTh 12:15 pm -1:30 pm, TuTh 3:00 pm - 5:45 pm, W 11:00 am - 3:00 pm.

Text: Walter Savitch, "Problem Solving with C++", ninth edition, Pearson, 2015.

Software: Feel free to use any C++ compiler you wish, including those available in the College's computer laboratories and classrooms, and those available for free downloading through UALR's Academic Alliance with Microsoft. The corequisite CPSC 1175 laboratory class will teach you how to do this.

Prerequisite: MATH 1302 (College Algebra) or higher mathematics course.
Corequisite: CPSC 1175 (Introduction to Computer Science -- Laboratory).
(Some prior familiarity with PC's, as would be gained in a computer literacy course, would be useful but is certainly *not* essential. The CPSC 1175 laboratory experience should suffice for this. No prior knowledge of computer programming is expected or assumed, but ability to think logically and algorithmically will be *highly* useful; MATH 1302 is a prerequisite for that reason, rather than specific knowledge of anything more than high-school algebra being required for the course.)

Grading: There will be seven programming assignments (for 40% of the total course grade -- each program will count 5%, with the last assignment counting as a "double" assignment worth 10%), and a midterm exam (20%) and final exam (40%). All exam and due dates will be announced well in advance. No make-up exams will be given; students who must miss an exam or an assignment due date should make **prior** arrangements with the instructor. Late homeworks will be accepted (with 1 point out of 10 lateness penalty) at the class meeting immediately following the due date, after which *they will not be accepted.* Furthermore, it is expected that all work submitted will be the result of the efforts of the individual submitting the work; any work which fails to meet this criterion *will not be accepted.* (I cannot overemphasize how important it is to write your own code; there is no other viable way to learn programming.)

Course description: The course essentially covers Chapters 1 through 6 (the basic procedural constructs of C++ programming), Chapter 7 (arrays), and parts of Chapters 8 (strings) and 10 (structures) of Savitch's text; it can be succinctly described as an introduction to elementary programming and problem-solving techniques via the C++ programming language. After a very brief survey of the history of technology and computing, the course will heavily focus on **programming**. Major topics will include: loop control, decision-making, input/output, files, constants and variables, character and numeric information, data types and expressions, strings, one- and multi-dimensional arrays, functions and procedures, and introduction to record structures. Throughout the course, the programming and problem-solving process will be emphasized: formulating algorithms, translating algorithms into code, and editing, compiling, linking, running, debugging, and refining code.

CPSC 1375 Programming I HW1

Write a C++ program which answers the following question:

How far do you have to go in adding together the sequence
1+2+3+4+5+6+...
for the sum to exceed 25,000?

That is, your program should keep adding together the above number sequence until the sum gets to be more than 25,000, at which point it should print out both the sum and the last number added which made the sum go over 25,000. Please print out appropriate messages with each of these, like:

The last number added was ________.
The final sum was _________.

Note that, since you have seen in class an easy way of adding arithmetic series, you can actually check the computer's work by hand here! (Something to think about: what might you do to convince yourself of your program's correctness if you didn't know how to sum up arithmetic series?)

Please hand in a paper copy of your C++ source code program (not the compiler listing) and a copy of your output. There are several possible ways of getting a printout of your output (depending on the environment in which you are working), such as screen dumping, logging the session, sending output directly to a printer, and writing output to a file for later printing. As this is your first assignment, and as the output is not voluminous, you can just copy the output by hand onto a sheet of paper and hand that in along with your source code.