

CS105 - Introduction to Computer Programming I

Last Updated - 01/25/02

Course Manager - Matthew Bauer, Senior Lecturer

2 credit hours; required for CS & CPE (or CS200); 100 min. lecture & 50 min. lab each week

Current Catalog Description - Introduces the use of a high-level programming language (C/C++) as a problem-solving tool—including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. (2-1-2)

Textbook

- Roberge/Bauer/Smith, *Engaged Learning for Programming in C++: A Laboratory Course*, Jones and Bartlett Publishers, 2nd Edition, ©2001, ISBN-0763714232

References - other textbooks or materials

- Deitel/Deitel, *C++ How To Program*, Prentice-Hall, Inc., 3rd Edition, ©2001, ISBN-01308957171

Course Goals - Students should be able to:

- Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions
- Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions
- Break a problem into logical pieces that can be solved (programmed) independently.
- Develop, and analyze, algorithms for solving simple problems.
- Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
- Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, and class libraries including strings and vectors

Prerequisites by Topic

- no prerequisites

Major Topics Covered in Course

1. Development Environment, C++ Program Elements	3 hours
2. Data Types, Expressions, Basic I/O, Data Type Conversion, Library Functions, Strings (introduction)	3 hours
3. Selection	6 hours
4. Stream File I/O, Output Manipulators	4 hours
5. Iteration	8 hours
6. Functions (scope, pass by reference, overloading)	3 hours
7. Arrays, Vector Class	9 hours
8. Project	5 hours
Quiz #1, Midterm Exam, Quiz #2	4 hours
Final Exam	-
	45 hours

Laboratory projects (specify number of weeks on each)

- 9 labs (1-2 labs each week, each lab contains multiple programming assignments, some with shells, and analysis work)
 - Programming using a Development Environment; C++ Program Elements I; C++ Program Elements II; Selection; File I/O and Streams; Iteration I; Iteration II; Functions I; Arrays and Using the Vector Class

- 1 procedural programming project (individual, 3 weeks, requiring at least 5 functions and use of class libraries, design and implementation)

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	.3		Computer Organization and Architecture	0	
Algorithms	.3		Concepts of Programming Languages	1	
Software Design	.3				

Oral and Written Communications - Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically _____ pages and to make 0 oral presentations of typically _____ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- Legitimate Code Re-Use, 1 hour, procedural programming project

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- none

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Most labs include problem solving with pseudo-code component, or debugging code segments, or determine program output.

Solution Design - Please describe the design experiences common to all course sections.

- 1 procedural programming project (individual, 3 weeks, requiring at least 5 functions and use of class libraries, design and implementation)

Other Course Information

- Additional Suggested Course Assignments
 - 2 programming quizzes (50 minutes each in lab)
 - 1 midterm exam (100 minutes, around 70% programming)
 - 1 final exam (120 minutes, around 70% programming)
- Planned Course Enhancements
 - Change catalog description to include objected-oriented approach. (Summer 2002)
 - Change to "objects-first" approach. (Fall 2002)
 - Stress problem solving, algorithms, and design more than programming language. (Fall 2002)