



CSC 115: Computer Science I: Introduction to C++ Programming

Course Description

CSC 115 Computer Science I is the first course in computer science for computer science majors and minors as well as science and mathematical majors. CSC 115 offers an overview of computer science and an introduction to programming. The course involves a high level programming language and the topics covered are data types, expressions, assignment, selection, repetition, and function. The course will develop and analyze algorithms for computational efficiency, the use of computers and graphing calculators to explore mathematical concepts, and develop computer programs in a structured language.

Instructional Emphasis: This course is more than just reading a textbook or answering multiple-choice questions. You must write programs in C++ in order to understand the syntax and semantics of this computer language. You will have many opportunities to practice in learning new techniques of writing programs using the C++ programming language. Computer programming projects and assignments will determine over 60 percent of your grade here. Programming in C++ is the emphasis of this course.

Course Objectives

The course will use computational tools and strategies to estimate properly, along with solving problems that arise in mathematics and as well as in topics outside of mathematics. CSC 115 offers a means to build mathematical knowledge through problem solving and organizing mathematical thoughts through communication, allowing you to create and use representations to organize, record, and communicate mathematical ideas. The course also allows you to select, apply, and translate among mathematical representations to solve problems.

Course Components

In this course, there are twenty-two graded assessments. I believe that you learn most by actually writing programs, not by memorizing phrases in textbooks. Accordingly, only 40 percent of your course grade will be based on test scores and textbook assignments. You will have three exams worth a total of 150 points, the mid-term exam worth 100 points, the final exam worth 100 points, the major course project worth 250 points, six computer programs that total 350 points, and homework assignments worth a total of 50 points, for a grand total of 1,000 points.

3 minor exams	150 points	15 percent
10 assignments from the textbook	50 points	5 percent
6 programming Assignments	350 points	35 percent
Final programming project	250 points	25 percent
Midterm exam	100 points	10 percent
Final exam	100 points	10 percent
TOTAL	1,000 points	100 percent

The letter grade is determined by dividing the total points by ten and is based on the usual ten-point scale: 90-100 (A), 80-89 (B), 70-79 (C), 60-69 (D), and 0-59 (F).

Required Materials

See course description for an up-to-date list of materials.

Honor Code

All work done in relation to this course is bound by the Honor Code. It is essential that you adhere to the principles outlined in the Honor Code at all times.

Course Mechanics

Submitting Assignments

Submit your work as an attachment to an e-mail message to **both your instructor and the Self-paced Courses office** by clicking a "Submit Assignment " button (located at the bottom of each lesson page). Clicking the button opens a pre-addressed e-mail to me and the Self-paced Courses office with the correct subject line. Add you name to the subject line. By providing the assignment number and your name, you make it easier for your instructor and Student Services to track your progress and keep your work separate from that of other students.

It is extremely important for you to save copies of any work you send via e-mail. If your instructor doesn't receive your message, you must have a copy of the e-mail with the attached file, indicating the date sent, to prove that you submitted the assignment on time. **It is your responsibility to maintain copies of your sent e-mails as well as the attached files**, as there is no way to guarantee that any e-mail message will be delivered.

Please check your e-mail software to see how it manages sent and saved messages. Some software automatically deletes messages one month after they have been sent; others only save messages if they are filed in folders; others save messages received but not those sent. You may need to send yourself a copy of your e-mailed assignment at the same time you send it to your instructor, or you may need to print a copy of the e-mail message and any attachments to keep in your paper files. No matter how your system works, make sure you know how to save a copy of all work that you submit

to your instructor and that you save the copy for several months beyond the end of the course.

Other Questions

Contact your instructor with questions regarding the content of the course and your progress. (There is an “Instructor’s e-mail” link at the top of every lesson page.) Please include “SPC CSC 115” in the subject line of your e-mail.

Contact the Instructional Designer at the Friday Center about problems with this Web site, including bad links.

If you have any logistical questions as you work through the course (enrollment, credits, withdrawal, and so on), contact the Student Services staff at the Friday Center for Continuing Education (phone 919-962-1134 or 800-862-5669).

Lessons

The course schedule is up to you. You can complete the course in as few as twelve weeks or take as long as nine months. The important thing is to get a good start, then maintain your momentum.

Since this course is not held in a classroom, I will probably never meet you in person. Still, I would like to know something about you so that I can associate each e-mail message from you with something more than a screen name. Therefore, your first task is to send me a Personal Information Sheet: Save this Word Document to your hard drive (or copy and paste from this Web page into your word processing program), fill it out, and attach it to an e-mail to me. This also gives us a chance to make sure our lines of communication are working.

Start by clicking the link below for Lesson 1.

Lesson 1	Chapter One: Getting Started Section 1.1: Introduction to Programming Section 1.2: Function and Class Names
Lesson 2	Section 1.3: The cout Object
Lesson 3	Section 1.4: Programming Style Section 1.5: Common Programming Errors Section 1.6: Chapter Summary
Lesson 4	Chapter Two: Data Types, Declarations, and Displays Section 2.1: Data Types
Lesson 5	Section 2.2: Arithmetic Operators Section 2.3: Numerical Output Using cout
Lesson 6	Section 2.4: Variables and Declarations Section 2.5: Common Programming Errors Section 2.6: Chapter Summary Section 2.7: Chapter Supplement (optional) Exam 1
Lesson 7	Chapter Three: Assignment and Interactive Input Section 3.1: Assignment Operations Section 3.2: Mathematical Library Functions Section 3.3: Interactive Keyboard Input

Lesson 8	Section 3.4: Symbolic Constants Section 3.5: Common Programming Errors Section 3.6: Chapter Summary
Lesson 9	Chapter Four: Selection Section 4.1: Relational Expressions Section 4.2: The if-else Statement Exam 2 Midterm
Lesson 10	Section 4.3: Nested if Statements Section 4.4: The switch Statement Section 4.5: Common Programming Errors Section 4.6: Chapter Summary Section 4.7: Chapter Supplement: Errors, Testing, and Debugging
Lesson 11	Chapter Five: Repetition Section 5.1: The while Statement Section 5.2: cin within a while Loop
Lesson 12	Section 5.3: The for Statement Section 5.4: The do Statement Section 5.5: Common Programming Errors Section 5.6: Chapter Summary
Lesson 13	Chapter Six: Modularity Using Functions Section 6.1: Function and Parameter Declarations Section 6.2: Returning a Single Value Section 6.3: Pass by Reference
Lesson 14	Section 6.4: Variable Scope Section 6.5: Variable Storage Class Section 6.6: Common Programming Errors Section 6.7: Chapter Summary Section 6.8: Chapter Supplement: Generating Random Numbers (Optional) Exam 3
Final Project	
Final Exam	Schedule your final exam.
Evaluation	Please submit a brief online course evaluation. Your opinion matters to us!