

# Course Syllabus – Spring 2009

## Catalog Description

Lecture, 2 hours; laboratory, 3 hours. Pointers and dynamic allocation of storage; linked lists; an introduction to the object oriented programming (OOP) paradigm; classes and objects; encapsulation; member variables and member functions; inheritance and polymorphism; scoping; templates; iterators; error handling techniques. Prerequisite: Completion of CS 110 and 115 with a C- or better, or consent of instructor.

This course is currently taught using C++.

## Prerequisites

CS 115 (CS 150) or instructor consent. Basic C/C++ programming skills are required. Students lacking a background in C/C++ will need to familiarize themselves with the language within the first two weeks of the course. Students who have not acquired a strong foundation in the fundamentals of procedural programming (i.e., earned a grade below a C- in CS 115) must retake CS 115 or its equivalent before taking this course.

## Instructor

Dr. Tia Watts Darwin 116E (707) 644-2807  
e-mail: [tiawatts@cs.sonoma.edu](mailto:tiawatts@cs.sonoma.edu) (Message subject MUST begin with **CS 215**)  
Office Hours: Tuesday 12:00 - 1:00 pm Office or Lab  
Thursday 2:00 - 3:00 pm Office or Lab

## Class Meetings

Lecture: Monday/Wednesday 2:30 – 3:45 pm Darwin 31  
Lab Section: Wednesday 10:00 am – 12:50 pm Darwin 28

## Course Objectives

Upon successful completion of this course, the student will be able to:

- Use predefined classes to declare objects in a C++ program.
- Design, implement, and use new C++ classes.
- Employ operator overloading in C++ classes.
- Design, implement, and use template classes.
- Use single and multi-dimensional arrays in a C++ class/program.
- Use static and dynamic arrays in a C++ class/program.
- Develop and implement recursive algorithms
- Design, implement, and use linked lists in a C++ class/program.
- Use container classes from the Standard Template Library (STL).
- Employ inheritance and polymorphism in C++ classes.
- Design and implement applications using a library of application development classes

## Important Dates

26 January 2009	First day of classes
6 February 2009	Last day to ADD or DROP courses
16 February 2009	President's Day Observed (No classes – University closed)
20 February 2009	Last day to WITHDRAW from courses online
4 March 2009	Midterm Exam 1
31 March 2009	Cesar Chavez Day Observed (No classes – University closed)
8 April 2009	Midterm Exam 2
13-17 April 2009	Spring Break (No classes – University open)
15 May 2009	Last day of classes
18 May 2009	Final Exam: 2:00 - 3:50 pm (tentative)

## Course Materials

### Text

Absolute C++; (3rd Edition)

by Walter Savitch; Addison Wesley Publishers; 200?

Getting Started with Microsoft Visual C++ 6 with an Introduction to MFC (2nd Edition)

by Harvey M. Deitel, Paul J. Deitel, Tem Nieto, and Edward Strassberger

### Other Materials

CS 215 Lab Manual

Loose Leaf Binder; Hole Punch; Small Stapler

## Coursework

### Lecture

The proposed outline of the topics to be covered appears in the course schedule. Students are expected to attend all lectures and to read the relevant sections of the text prior to lecture. Students are responsible for making up the missed work if they are absent. *All electronic devices must be turned off during lecture.*

### Labs

The lab meetings provide an opportunity for students in the class to interact with each other and with the instructor, to get immediate solutions to coding problems, and to get advice on the process of programming. The labs for this course are designed to give the student an opportunity to work with a variety of programming concepts. Lab attendance is essentially mandatory. Lab exercises are due at the end of the lab period; an extension (until the beginning of the next lab) will automatically be given *only* to students who have not completed the assignment but have attended the full lab.

### Programming Projects

Several programming projects will be assigned during the semester. Each project will require that you write one or more C++ program modules. Project due dates and times will be stated on the project assignment handout. Projects must be submitted by the due date and time. Submission times will be based on the UNIX date/time stamp on the file(s). No projects will be accepted after the due date and time.

### Midterms and Final Exam

The examinations are based on the assumption that each student is responsible for knowing the material covered in class (lecture and lab) and also material covered in relevant sections of the course text. All exams are closed book; however, one 8 ½ by 11 *handwritten* sheet of notes (front and back) will be allowed. The final exam will be comprehensive. There may be one or more unannounced quizzes given during the semester. Exams cannot be made up unless you have made arrangements with the instructor prior to the date of the exam. Quizzes cannot be made up.

## Grading

Homework, Lab Attendance and Exercises	20%
Programming Project	30%
Exams and Quizzes	50%

### Grading Scale:

100	...	93	...	90	...	87	...	83	...	80	...	77	...	73	...	70	...	67	...	63	...	60	...	0
	A		A-		B+		B		B-		C+		C		C-		D+		D		D-		F	

Note: You must separately earn a passing grade on the programming projects and the exams in order to pass the course.

CS Majors must take this course for a letter grade. University guidelines regarding the grade of Incomplete will be strictly adhered to. Incomplete grades will only be given for circumstances beyond a student's control; inability to keep up with the work due to an excessive course load, for example, is insufficient to warrant an Incomplete. The University also requires that, to be a candidate for an Incomplete grade, the student must currently be doing C (or better) work in the course.