

## COURSE DESCRIPTION

**Department and Course Number:** CSCI 390

**Course Title:** Special Topics in Programming  
(Topic described is Object-Oriented Programming using C+.)

**Current Catalog Description:** Study of topics in programming according to the interests of instructor and students.

**Total Credits:** 3 hours

**Coordinator:** H. Conrad Cunningham, Chair & Associate Professor of Computer and Information Science. Developer of Object-Oriented Programming using C++ topic is Jason Hale, part-time instructor.

**Textbooks:** Arthur J. Riel, *Object-Oriented Design Heuristics*, Addison Wesley, 1996.  
H. M. Deitel and P. J. Deitel. *C++ How to Program*, Third Edition, Prentice Hall, 2001.

**References:** Andrew C. Staugaard, Jr. *Structured and Object-Oriented Problem Solving Using C++*, Third Edition, Prentice Hall, 2002.

Other materials indexed at <http://www.mcsr.olemiss.edu/bookshelf/doc/c/resources.html>

**Course Goals:** Students who successfully complete this course will be able to design and implement C++ programs to solve general programming problems. They will be able to identify and understand the concepts, terminology, and methods of the object-oriented approach and have a good understanding of its advantages and disadvantages with respect to the action-oriented approach. Students will complete a group programming project demonstrating their abilities to collaboratively arrive at and implement a reasonable object model for solving a programming problem.

### Prerequisites by Topic:

1. Fundamental programming concepts and skills (CSCI 111, 112, 211)
2. Basic data structures and algorithms (CSCI 112, 211)

### Major Topics Covered in the Course:

1. Motivation for structured and object-oriented programming (3 hours)
2. Classes and objects (3 hours)
3. Differences between action-oriented and object-oriented programming (3 hours)
4. Uses relationship (2 hours)
5. Containment relationship (2 hours)
6. Inheritance (4 hours)
7. Polymorphism (3 hours)
8. Multiple inheritance (1 hour)
9. Fundamental C and C++ constructs and language features (3 hours)
10. Iteration and decision structures (1 hour)
11. Function overloading (1 hour)
12. Separate compilation (1 hour)
13. Using standard library functions and the Standard Template Library (1 hour)
14. Object and class syntax in C++, including single/multiple inheritance, containment, constructors, destructors (3 hours)
15. Operator overloading (1 hour)
16. Identifying and refining object models from problem requirements (6 hours)
17. Examinations (3 hours)

**Laboratory Projects:**

1. Program to demonstrate competence in basic C structures
2. Program to demonstrate competence representing a problem as objects, using function overloading
3. Program to demonstrate competence using containment/inheritance/polymorphism/operator overloading with objects
4. Project to demonstrate ability to collaboratively arrive at and implement a reasonable object mode to represent a problem

**Estimate of ABET/CAC Category Content:**

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

**Oral and Written Communications:** Every student is required to make one oral presentation of typically five minutes in duration during the semester.

**Social and Ethical Issues:** The class has one quiz covering the School of Engineering Honor Code. The class also discusses the ethical issues of deadline and requirements management.

**Problem Analysis:** Students must analyze problems to devise appropriate programming solutions. There are class exercises to prepare the students for this analysis.

**Solution Design:** Students must design their own final solutions (code) to the programming assignments, except for the group project. Early in the semester, the instructor guides the students' designs of the class structures needed in the assignments. As the students gain experience in use of object-oriented techniques in C++, they are expected to do progressively more of the design themselves.