

COURSE DESCRIPTION

Department and Course Number	CS 302	Course Coordinator	Ranelli
Course Title	Object-Oriented Design	Total Credits	4

Current Catalog Description

CS 302 is a continuation of CS 201 and emphasizes concepts of object oriented software design. Topics include inheritance, recursion algorithm analysis, sorting algorithms, graphs, hash tables, and linked list data structures such as stacks, queues, and binary trees. This class has a laboratory component.

Textbook

Data Structures and Problem Solving using Java, 2nd ed., by Mark Allen Weiss, Addison-Wesley, 2002.

References

Java Software and Solutions: Foundations of Program Design, 5th ed., by John Lewis and William Loftus, Addison-Wesley, 2007.

Course Goals

Students will acquire knowledge and skills in the following areas:

Inheritance

The comparison of iterative and recursive algorithms and the analysis of their time complexity using big-Oh analysis.

Object oriented program design using data structures composed of objects. Specific data structures include arrays, linked lists, array and link based stacks and queues, binary trees, binary search trees, expression trees, general graphs and hash tables.

The design and analysis of algorithms using the above listed data structures, such as: data insertion and removal, searching in array and linked list structures, Huffman codes (binary trees) weighted, unweighted shortest paths in general graphs

An introduction to the analysis and use of sorting algorithms such as insertion sort, Shell sort, quick sort, and merge sort.

Prerequisites

Introduction to Object-Oriented Programming with “C” or better

Major Topics Covered in the Course

See Course Goals above

Laboratory projects (specify number of weeks on each)

There are 15 weekly lab sessions, including two lab programming exams.

There are five homework programming assignments, approximately two weeks is allowed for each assignment.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	<u>20</u>	<u>0</u>	Computer Organization and Architecture	<u>0</u>	<u>0</u>
Algorithms			Concepts of Programming Languages		
Software Design	<u>20</u>	<u>0</u>		<u>0</u>	<u>0</u>

Oral and Written Communications

None

Social and Ethical Issues

None

Theoretical Content

Analysis of Algorithms (5 hours)

Problem Analysis

A large component of this course is the design and analysis of algorithms using recursion and data structures implemented as objects. Students are assessed using five homework programming assignments, two lab exams, and three class exams.

Solution Design

See Problem Analysis above.