

CSE 2222 (Approved): Software I.5-Transition: Development of Software Components

Course Description

Templates for generalization and decoupling; container components; component-based software from implementer's perspective; data representation using layering and using pointers. Transition course: requires CSE 221, serves as prereq for CSE 2231.01.

Prior Course Number: CSE 222

Transcript Abbreviation: SW1.5: Dev SW Cpts

Grading Plan: Letter Grade

Course Deliveries: Classroom

Course Levels: Undergrad

Student Ranks: Freshman, Sophomore

Course Offerings: Summer

Flex Scheduled Course: Never

Course Frequency: Every Year

Course Length: 7 Week

Credits: 3.0

Repeatable: No

Time Distribution: 4.0 hr Lec, 2.0 hr Lab

Expected out-of-class hours per week: 12.0

Graded Component: Lecture

Credit by Examination: No

Admission Condition: No

Off Campus: Never

Campus Locations: Columbus

Prerequisites and Co-requisites: CSE 221

Exclusions: Not open to students with credit for CSE 222

Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: Yes

The course is a GEC: No

The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901

Subsidy Level: Baccalaureate Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering

General Information

RESOLVE/C++ is used
This course is equivalent to CSE 222 and will be offered ONLY in Su 2012 for students who have taken CSE 221 in Sp 2012 (or earlier) and have not yet taken CSE 222. It will prepare them to take 2231.01 which will be offered ONLY in Au 2012.

Course Goals

Be competent with using the computing environment (operating system, tools, language system, etc.) to complete lab assignments.
Be competent with using C++ class templates and classes, and related RESOLVE/C++ principles for clients, to write layered implementations of operations.
Be competent with using simple recursion to write layered implementations of operations.
Be competent with using simple techniques to test layered implementations of operations, including developing and carrying out simple specification-based test plans.
Be competent with using simple techniques to debug layered implementations of operations.
Be familiar with using the Array, Binary_Tree, List, Partial_Map, Queue, Record, Representation, Sequence, Set, Sorting_Machine, and Stack templates to write application programs and/or component implementations.
Be familiar with using basic C++ control structures and statements, RESOLVE/C++ class templates and classes, and related RESOLVE/C++ principles for clients and implementers, to write component realizations with layered data representations.
Be familiar with writing code that uses dynamic storage management and pointers for components that define new types with simple "raw C++" linked data representations.
Be familiar with using basic algorithm analysis techniques and notations to analyze and express execution time of operations whose implementations involve straight-line code and simple loops.
Be familiar with using simple formal logic assertions involving mathematical set models to understand and reason about an operation's behavior.
Be familiar with using simple techniques to test implementations of class templates that define new types, including developing and carrying out simple specification-based test plans.
Be familiar with using simple techniques to debug implementations of class templates that define new types.
Be exposed to using data representation conventions ("representation invariants") and correspondences ("abstraction relations") to reason about correctness of data representations.
Be exposed to using the RESOLVE/C++ principles for interface designers to guide the choice of mathematical model and operation behavior of a new software component.
Be exposed to using loop invariants to reason about loop behavior.

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Review of generalization using templates and template instantiation	3.0		1.0					
Decoupling using templates; utility classes	3.0		1.0					
How recursion works; Partial_Map and its many uses	3.0		1.0					
Implementer's view; data representation using other objects; representation invariant and abstraction relation; commutative diagram and correctness of data representation	6.0		2.0					
Hashing	3.0		1.0					
Binary trees; binary search trees	3.0		1.0					
Pointers and dynamic storage allocation and deallocation; singly linked and doubly linked list data structures	6.0		2.0					
Review and exams	3.0							

Representative Assignments

E-mail classifier
Automated glossary generation
Partial_Map represented using Array of Queues of Records and hashing
Partial_Map represented using binary search tree
List represented using "raw C++" pointers

Grades

Aspect	Percent
Class Participation	2%
Closed Labs	8%
Programming Lab Assignments	30%
Homework Assignments	10%
Midterm Exam	20%
Final Exam	30%

Representative Textbooks and Other Course Materials

Title	Author
<i>Software Component Engineering with Resolve/C++, volumes 2 and 3</i>	Weide, B.W., OSU UniPrint, 2007

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
***	a	An ability to apply knowledge of mathematics, science, and engineering.
*	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
***	c	An ability to design a system, component, or process to meet desired needs.
**	d	An ability to function on multi-disciplinary teams.
**	e	An ability to identify, formulate, and solve engineering problems.
*	f	An understanding of professional and ethical responsibility.
**	g	An ability to communicate effectively.
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
*	i	A recognition of the need for, and an ability to engage in life-long learning.
	j	A knowledge of contemporary issues.
***	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
***	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
*	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
***	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
**	d	an ability to function on multi-disciplinary teams;
**	e	an ability to identify, formulate, and solve engineering problems;
*	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
**	g	an ability to communicate effectively with a range of audiences;

Course Contribution		Program Outcome
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
*	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
	j	a knowledge of contemporary issues;
***	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
**	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
**	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
***	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Additional Notes or Comments

This is a "bridge" course that will get students who have completed CSE 221 by the end of Sp '12 ready to take the CSE 2231.01 in Au '12. Together, CSE 221, this course (2222), and 2231.01 will be equivalent to Software I, II.

Sunset: This course will only be offered in Su 12.

Prepared by: Neelam Soundarajan