

CSE 2232 (Approved): Software II.5-Transition: Software Development in Java

Course Description

Transition from quarters to semesters, Resolve/C++ to Java; introduction to Java; tools for coding, testing, version control, documentation; language-specific best practices stemming from principles of component-based design.

Prior Course Number: 421

Transcript Abbreviation: SW II.5: Dev & Dsg

Grading Plan: Letter Grade

Course Deliveries: Classroom

Course Levels: Undergrad

Student Ranks: Sophomore

Course Offerings: Autumn

Flex Scheduled Course: Never

Course Frequency: Every Year

Course Length: 14 Week

Credits: 2.0

Repeatable: No

Time Distribution: 2.0 hr Lec

Expected out-of-class hours per week: 4.0

Graded Component: Lecture

Credit by Examination: No

Admission Condition: No

Off Campus: Never

Campus Locations: Columbus

Prerequisites and Co-requisites: CSE 321

Exclusions: Not open to students with credit for CSE 421 or CSE 2221 or CSE 2222 or CSE 2231 or CSE 2231.01

Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: No

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

This is a transition course. It is intended for students who have completed CSE 321 but not 421. Under semesters, this material is covered in CSE 2221 and 2231.

Course Goals

Master core Java language features including: objects, classes, interfaces, inheritance, and exceptions
Be competent with core SDK packages including: collections framework, logging, and IO
Be competent with core best practices for component-based development including: separation of abstract state and concrete representation and coding to the interface
Be competent with the use of a modern IDE, such as Eclipse
Be familiar with advanced language features including: iterators, generics, and assertions
Be familiar with foundations of an object-oriented paradigm, in particular: encapsulation, inheritance, and polymorphism
Be familiar with the application of design patterns including: immutable objects, factories, and singleton objects
Be familiar with best practices with regards to object equality, object cloning, and checked/unchecked exceptions
Be familiar with CVS, JUnit, and Javadoc
Be exposed to advanced SDK packages including: Swing for GUIs and network programming
Be exposed to exotic language features including: nested classes, nested interfaces, and annotations

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Overview: compilation, primitive types, reference semantics	3.0							
Language: Objects and classes, packages, generics, inheritance, interfaces, exceptions, reflection, garbage collection, nested classes, annotations	9.0							
Packages: Collections, Logging, IO, Swing, Network Programming	5.0							
Best Practices: equality, cloning, immutable objects, exceptions	5.0							
Patterns: factories, singletons	2.0							
Tools: Eclipse, CVS, Junit, Javadoc	6.0							

Representative Assignments

Implementing, testing, and documenting an arbitrary-sized natural number type
GUI for data mining using Swing
N-gram generator using the collections framework

Grades

Aspect	Percent
Individual assignments	50%
Team assignments	20%
Exams	25%
Participation	5%

Representative Textbooks and Other Course Materials

Title	Author
<i>Core Java, Volume 1</i>	Horstmann and Cornell

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;
	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.

Prepared by: Paolo Sivilotti