

# Course Information Sheet

## CSCI 1302

### Software Development

#### Brief Course Description (50-words or less)

Software development techniques in an object-oriented computer language. An intermediate programming course emphasizing systems methods, top-down design, testing, modularity, and structured techniques. Applications from areas of numeric and non-numeric processing and data structures.

#### Extended Course Description / Comments

This course is the 2<sup>nd</sup> in a 2-part series of courses introducing students to the Java programming language. This course includes group work and/or pair-programming.

As per the CSCI Department's requirements, all coursework in major-related courses and pre-requisite courses needs a grade of "C" or higher.

#### Pre-Requisites and/or Co-Requisites

This class may be taken in parallel to CSCI 1730 or as a CSCI 1302 -> CSCI 1730 sequence.

CSCI 1301

Introduction to Computing and Programming

Author(s): Lewis, DePasquale, Chase

Title: *Java Foundations: Introduction to Program Design & Data Structures*

Edition: 2<sup>nd</sup> Edition

ISBN-13: 978-0-13-212881-0

Author(s): Savitch

Title: *Java: An Introduction to Problem Solving and Programming*

Edition: 5<sup>th</sup> Edition

ISBN-13: 0-13-607214-3

**Approved Textbooks**  
(if more than one listed, the textbook used is up to the instructor's discretion)

#### Learning Outcomes (Performance Indicators)

This course presents software development techniques in the Java programming language. At the end of the semester, all students will be able to do the following:

1. Plan, design, implement, test and debug, and deploy a complete object-oriented software solution in Linux/Unix environment
2. Define recursion and apply it as a problem-solving technique
3. Define inheritance and apply it in a software project
4. Define polymorphism and apply it in a software project
5. Classify and use common data structures including lists, queues, arrays, and stacks
6. Assess a Java method's time and space complexity
7. Apply Swing components in creating a graphical user interface
8. Implement exception-handling in software projects and explain the importance of doing so
9. Describe techniques in software design including design patterns, unit testing, and debugging techniques
10. Apply pair-programming principles on a software-based project

**Outcomes and Student Outcomes**

	a	b	c	d	e	f	g	h	i	j	k	
<i>Learning Outcomes</i>	1	•	•	•	•				•	•	•	
	2	•		•					•	•	•	
	3	•		•					•			
	4	•		•					•			
	5	•	•	•					•	•		
	6	•	•								•	
	7	•	•	•					•			
	8	•	•	•		•				•		
	9	•		•		•				•		•
	10				•					•		

**Student Outcomes**

- a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
- b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- d. An ability to function effectively on teams to accomplish a common goal.
- e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
- f. An ability to communicate effectively with a range of audiences.
- g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
- h. Recognition of the need for and an ability to engage in continuing professional development.
- i. An ability to use current techniques, skills, and tools necessary for computing practice.
- j. 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.
- k. An ability to apply design and development principles in the construction of software systems of varying complexity.

**Major Topics Covered**

(Approximate Course Hours)

3 credit hours = 37.5 contact hours

4 credit hours = 50 contact hours

Note: Exams count as a major topic covered

Programming Fundamentals	(14 hours)
Data Structures Applications	4 hours
Recursion	2 hours
Event-driven	4 hours
Object-Oriented	5 hours
Algorithms and Complexity	(2 hours)
Sorting/Searching	2 hours
Human-Computer Interaction	(5 hours)
HCI Foundations	3 hours
Building GUIs	2 hours
Graphics and Visual Computing	(1 hour)
Model-View Controller	1 hour
Social and Professional Issues	(1 hour)
Risks	1 hour

Software Engineering	(18 hours)
Using APIs, JavaDoc	2 hours
Software Design	8 hours
Requirements/Specs	1 hour
Project Management	3 hour
UNIX environment	3 hours

Exams (2 hours)

Miscellaneous (4 hours)

Special Topics (3 hours)

**Assessment Plan for this Course**

Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and results returned to the instructor of the course.

The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.

**How Data is Used to Assess Program Outcomes**

Each course Learning Outcome, listed above, directly supports one or more of the Student Outcomes, as is listed in "Relationships between Learning Outcomes and Student Outcomes". For CSCI 1302, Program Outcomes (a), (b), (c), (d), (e), (i), (j), (k) are supported.

**Course Master**

Dr. Plaue

**Course History**

1998 Fall Course Proposed  
2012-Feb-1 Course Information Updated