



CSc 020: Programming Concepts and Methodology II

- ❑ Instructor: Doan Nguyen, Ph.D.
- ❑ Lab Instructor: Abida Mukarram, Ph.D.
(lab sections: 2,3, 12, and 13 only)
- ❑ Office: ECS 3002
- ❑ E-mail: doan.nguyen@csus.edu
- ❑ Web page: <http://athena.ecs.csus.edu/~nguyendh>
- ❑ Office Hours: 10:30AM-12:00 PM – T Th.

Course Catalog Description

- ❑ Application of object-oriented techniques for systematic problem analysis and specification, design, coding, testing, and documentation.
- ❑ Semester-long project approach emphasizing larger programs.
- ❑ Managing program complexity using abstraction.
- ❑ Introduction to algorithm analysis and Big-O notation.
- ❑ Advanced language features.
- ❑ Basic sorting and searching algorithms.
- ❑ Recursion.

Course Goals

- ❑ Experience designing and developing large, complex programs in the context of a semester-long project.
- ❑ Use of object-oriented techniques to foster abstraction.
- ❑ A disciplined approach to the design, coding, testing and debugging of programs.
- ❑ Use of appropriate tools and techniques for each step of program development.
- ❑ Illustration of the power of abstraction via the use of multiple representations of linear lists.
- ❑ Reinforcement and expansion of object-oriented skills.
- ❑ Basic sorting and search algorithms and their analysis.

Course Is Not Just About Java

- ❑ May seem to focus on Java
 - All programming in Java
 - Many advanced Java language features
- ❑ Lessons intended to be general
 - Principles should apply to all OOP languages
 - ✓ Ways of thinking about design
 - ✓ General ideas about software
 - Can translate skills to other languages

Assume You Already Know

❑ Coding

- Variables, operators, loops, 1-D arrays

❑ Basic object-oriented programming

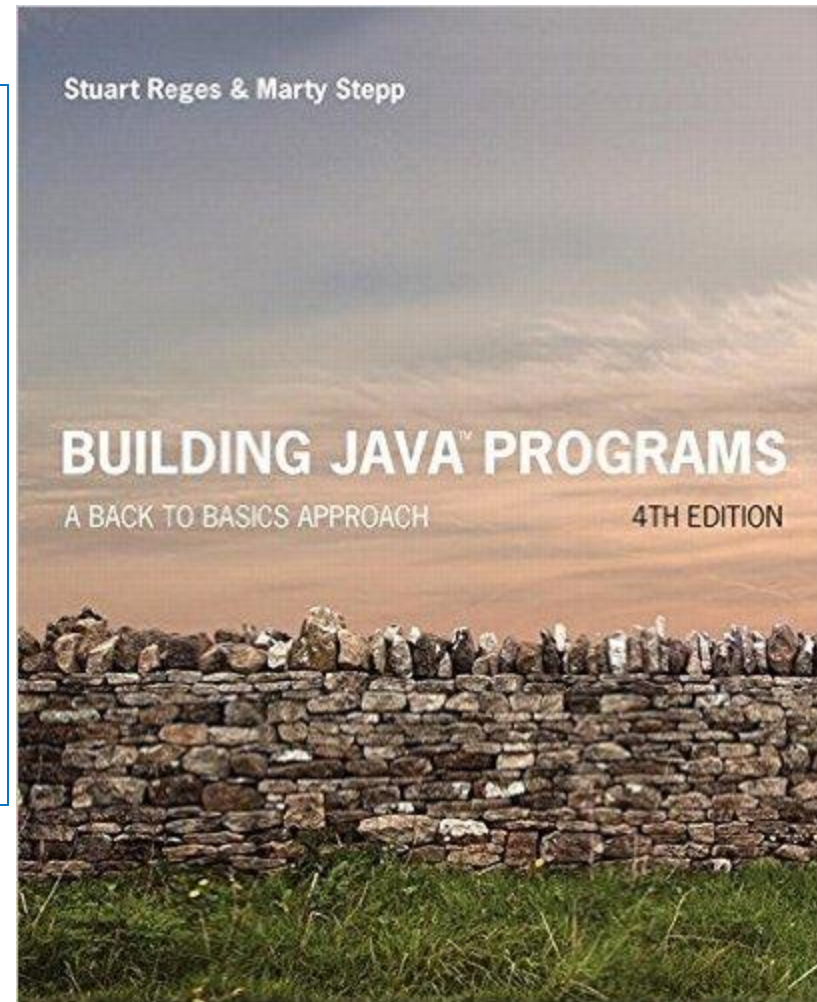
- Classes, members, methods, constructors

❑ Java

- Class libraries

Textbook

- ❑ Building Java Programs: A Back to Basics Approach, 4/E.
- ❑ By Stuart Reges and Marty Stepp.



Grading and Evaluation

Grades will be based on:

- I. Attendance and participation 15%
- II. Programming assignments and exercises
 - a) Labs 35%
 - b) Homework 5%
- III. Examinations
 - a) Midterm 20%
 - b) Final examination 25%

Class Notes

- ❑ To pass the course, you need passing grades for I, II & III.
- ❑ Examinations cannot be made up.
- ❑ To turn in a project/lab you need to
 1. Attach a softcopy of the full documented source code and its output to submit to LMS (Canvas). (Emailing your work will NOT be accepted!)
 2. Do a demo.
- ❑ Late work will not be accepted.
- ❑ Be aware of the school's policy on drops and incompletes.

Communication

- ❑ **LMS (Canvas): canvas.csus.edu**
 - ❑ assignments
 - ❑ announcements (via LMS (Canvas) with emails)
 - ❑ feedback and grades
- ❑ **Check your SacLink email and LMS (Canvas) daily**

Academic Honesty

- ❑ All projects, labs & exams must be done individually.
- ❑ Do not copy (or allow others to copy) your work in any way.
- ❑ Submissions will be compared to submissions from current and previous semesters.
- ❑ Cases of academic dishonesty will be referred to the Office of the Vice President for Student Affairs
- ❑ Visit <http://www.ecs.csus.edu/wcm/csc/academic/academicintegrity.html> for more detailed explanation of academic dishonesty.

Tentative schedule

Tentative Schedule:

Week	Lecture Topics	Lab Assignments
1	Class Information Java Basic	Introduction. Lab Info.
2	Java statements	Introductory Lab (warming up)
3	Coding design, style, doc. & optimization, Java Classes	Lab 01
4	Java arrays and Strings	Lab 02
5	Object Oriented Analysis and Design, Java Classes	Lab 03
6	List Abstract Data Type (ADT) Testing and Debugging, Midterm Review	Lab 04 (2 weeks)
7	List ADT (continue) Midterm examination	
8	Stack and Queue	Lab 05
9	File I/O, Polymorphism	Lab 06
10	Abstract Classes & Interfaces	Lab 07
11	Sorting & Searching	Final Project (2 weeks)
12	Introduction to Analysis of Algorithms	
13	Recursion	Final examination preparation
14	Java Generic & Exception handling in Java	
15	Course Review, Study Guides, Introduction to other related CS Courses	



Course Advice

- ☐ Start labs/projects **early**
- ☐ Ask questions
- ☐ Read the textbook
- ☐ Attend lectures
- ☐ Attend labs
- ☐ Use the instructor's office hours

CSC 20 resources (I need help!)

- ❑ Visit Instructor Office Hours (T/Th: 10:30AM-12 noon)
- ❑ Utilize demonstration source codes (to be supplied along with lecture notes)
- ❑ Visit Tutor Center: (Old schedule – Fall 2017)

Computer Science Tutoring

Times:	Monday	Tuesday	Wednesday	Thursday	Friday
9-10	Justin Selke (9-2)	Matthew Roy (9-10)	Justin Selke (9-1)	Matthew Roy (9-10)	
10-11					
11-12					
12-1					
1-2			Daniel Rudy (1-3)		
2-3	Daniel Rudy (2-3)	Matthew Roy (2-4)		Matthew Roy (2-4)	
3-4					
4-5					
5-6					
6-7					

(Check http://www.ecs.csus.edu/wcm/student_resources/ECS_tutoring.html for new schedule or contact the department)

Questions?