

CpS 209 Object-Oriented Programming II 01 Spring 2025
College of Arts and Science

Instructor: Sarah Gothard, Ph.D.

Email: sgothard@bju.edu

Office: Alumni 84

Telephone: (864) 242-5100 ext. 8152
(937) 321-5167 for urgent texts

Office Hours: MWThF 8:15-9:45 am
MWF 12:00-12:45 pm *by appointment only*
T 8:15-10:30 am
TTh 2:00-2:45 pm

Course Information

Introduces fundamental concepts needed to support the development of desktop applications. Topics include GUI frameworks, object-oriented design with design patterns, model-view architecture, introductory generic programming, and functional programming techniques. *Prerequisite: CpS 110.*

Program Learning Outcomes (PLO):

- Write, debug, and test programs using the object-oriented paradigm
- Describe and apply standard object-oriented Design Patterns
- Develop graphical event-driven programs using a professional IDE and GUI framework

Course Resources

Website: Please keep up with the course page at <https://bju.instructure.com/>

Textbook: [The Java Workshop](#)

Grading

Qty	Item	Points	Total	Scale:
2	Quizzes	10	20	A 90-100%
10	Labs	10	100	B 80-89%
5	Programs	70	350	C 70-79%
1	Team Project	120	120	D 60-69%
2	Lab Tests	70	140	F <60%
3	Written Tests	80	240	

Total Points: 950

Programming Assignments: There are two types of assignments in the class: labs and programs. Labs are small-scale assignments that typically take an hour or two to complete. Programs are larger-scale assignments that will likely take many hours (10-15 or more). Programs are graded as follows:

- **60% Correctness:** Program produces correct results; runs according to specification. Attention to the program specification is very important here.
- **20% Style:** Code is written according to style guidelines and instructor's design requirements. Consistency and attention to detail are important. The goal is clear, easily understandable code, thoughtfully commented.
- **15% Reports:** Each program and project submission must be accompanied by a written report.
- **5% Submission:** Program assignment submitted according to instructions.

Course Policies

In this course, topics build on the previous topic. Thus, if you fall behind, you will struggle with new content. For this reason, I do not accept late work. Work is due at the deadline. **Late work receives a 0.** Extensions may be purchased with [tokens](#).

Do not share class notes with anyone who is not enrolled in the same class section as you are during the same semester.

Professionalism

University classes are a place to sharpen your professional habits. Arrive on time. Dress appropriately. Engage with the material. Take pride in your work. Build relationships. Encourage growth in others.

University Policies

Handbook Policies

Compliance with student handbook policies is expected during class.

Attendance Policy

You are expected to attend class and be on time: <https://home.bju.edu/bju-policies/>. A partial attendance will be recorded when you miss the beginning or end of a class. If you miss more than 15 minutes of class, you will be marked absent. Students who exceed the allowed absences may be withdrawn from class.

If you need to miss class any reason, please contact me as soon as possible. Assignments and tests should be completed before planned absences.

Accommodations for Students with Disabilities

Students are required under Section 504 to communicate the need for accommodations and provide documentation to the Academic Resource Center Accommodations Office in AL 213. Visit <https://success.bju.edu/> for more information. Students are encouraged to seek an appointment in the first week, as accommodations are not provided retroactively.

Academic Honesty and Integrity Policy

See the Computer Science Department's Academic Integrity Policy:

<https://cs.bju.edu/academics/policies/academic-integrity-policy/>

Taking credit for someone else's work is unethical in any setting. In a university setting, it undermines the ability of faculty to accurately evaluate your competence, harming you and the reputation of the department. For these reasons, the penalties for academic dishonesty may be severe.

Generative AI

Since the goal of the assignments in this course is to learn to develop the skills covered NOT complete the tasks assigned, and since the use of AI to complete or jumpstart tasks defeats the goal of the assignments, you may not use generative AI tools (i.e. Chat GPT, Bing Chat, Google Bard, etc.) in this course for any assignment without the professors express permission. Should an AI tool be used with permission, its use must be documented.

© Copyright 2025 Sarah Gothard. Students are prohibited from disseminating notes to anyone outside of this section this semester. Students are prohibited from distributing other course materials. This course uses copyrighted material from other sources. This material is being used by permission.

Curriculum Information

Context

This course supports the following objectives of the Computer Science and Information Technologies programs:

CS 1. Design and implement solutions to practical problems

CS 8. Demonstrate understanding of fundamental concepts in the student's discipline

Learning Objectives

Objective	Content	Assessment
Write, debug, and test programs using the object-oriented paradigm (CS 1)		Programs 1-5 Lab Tests 1, 2 Test 1
Describe and apply standard object-oriented Design Patterns (CS 8)		Programs 3, 4; Test 3

Develop graphical event-driven programs using a professional IDE and GUI framework (CS 1)

Programs 1-5; Test 2

Tentative Schedule

Day	Topic	Text/Asgs	Due
Wed, Jan 15	Java Fundamentals		
Thu, Jan 16	Lab 1		
Fri, Jan 17	Classes, Gradle	Chapter 1: Getting Started	
Mon, Jan 20	<i>MLKj Day, no class</i>	How to Videos	
Wed, Jan 22	Unit Testing, More Java; Program 1	Chapter 2: Basics	
Thu, Jan 23	Lab 2		Lab 1
Fri, Jan 24	Arrays, Lists, Factories	Chapter 3: OOP	
Mon, Jan 27	Strings		
Wed, Jan 29	Test 1 topics , File formats, PNG Format, File IO	Chapter 4: Collections	Quiz 1 Quiz 2
Thu, Jan 30	Lab 3		Lab 2
Fri, Jan 31	Program 2, Design a Reader, OO Design	Chapter 5: Exceptions	Program 1
Mon, Feb 03	Test 1		
Wed, Feb 05	GUI Apps, Interfaces		
Thu, Feb 06	Lab 4		Lab 3
Fri, Feb 07	TDD, Model-View Architecture		
Mon, Feb 10	GUI Programming, Test 2 topics		
Wed, Feb 12	Inheritance		
Thu, Feb 13	Lab 5		Lab 4
Fri, Feb 14	Class Hierarchies		Program 2

Mon, Feb 17	Test 2	Lab Test 1 Practice	
Wed, Feb 19	<i>Bible Conference</i>		
Thu, Feb 20	<i>Bible Conference</i>		
Fri, Feb 21	<i>Bible Conference</i>		
Mon, Feb 24	Even More Java		
Wed, Feb 26	Dynamic Controls		
Thu, Feb 27	Lab Test 1		Lab 5
Fri, Feb 28	Anonymous Methods	Chapter 13: Function Programming	Program 3
Mon, Mar 03	Observer Pattern		
Wed, Mar 05	Timers, Animation, Sound		
Thu, Mar 06	Lab 6		
Fri, Mar 07	Intro to Git, Functional Programming	Chapter 15: Streams	
Mon, Mar 10	Generics, Choose teams		
Wed, Mar 12	Command Pattern, Project Introduction	Lab Test 2 Practice Download Lab Test 2 Practice	Program 4
Thu, Mar 13	Lab: Proposal		
Fri, Mar 14	Test 3 Topics, State Pattern, State Machines	Command Pattern	
Mon, Mar 17	Git, Data binding		
Wed, Mar 19	Test 3		
Thu, Mar 20	Lab 7		Lab 6
Fri, Mar 21	Project Proposal Presentations		Project Proposal

			Presentations Proposal
Mon, Mar 24	<i>Spring break</i>		
Wed, Mar 26	<i>Spring break</i>		
Thu, Mar 27	<i>Spring break</i>		
Fri, Mar 28	<i>Spring break</i>		
Mon, Mar 31	Serialization		
Wed, Apr 02	Workday		Program 5: Wednesday
Thu, Apr 03	Lab Test 2		
Fri, Apr 04	Design Patterns		
Mon, Apr 07	Dependency Inversion		
Wed, Apr 09	Workday		1st Sprint
Thu, Apr 10	Lab 8: Git		Lab 7
Fri, Apr 11	Workday		
Mon, Apr 14	Multithreading		
Wed, Apr 16	Multithreading with Servers		
Thu, Apr 17	Lab 9		Lab 8: Git
Fri, Apr 18	Asynchronous Methods		2nd Sprint
Mon, Apr 21	Workday		
Wed, Apr 23	Workday		
Thu, Apr 24	Lab 10		Lab 9
Fri, Apr 25	Workday		
Mon, Apr 28	Workday		

Wed, Apr 30	Final Exam Topics , Workday		
Thu, May 01	Workday, Lab 11		Lab 10 ; Final Sprint
Fri, May 02	Project Demos		Lab 11 ; Demos
Wed, May 07	3:30-4:40 p.m. Final		