

# CPS 110: Introduction to Object-Oriented Programming

Instructor / 191-3 Lab Instructor	Ethan McGee
Office	AL 70 (3rd Floor Left Hall)
Email	<a href="mailto:emcgee@bju.edu">emcgee@bju.edu</a>
Phone	864.201.2635 (texting acceptable)
Office Hours	<b>MWF:</b> 1:00 - 1:50 <b>Th:</b> 10:00 - 10:50 <b>T:</b> electronic only

191-1 / 191-2 Lab Instructor	Jim Knisely
Office	AL 64 (3rd Floor Left Hall)
Email	<a href="mailto:jknisely@bju.edu">jknisely@bju.edu</a>
Office Hours	

Text(s)	<i>How to Think Like a Computer Scientist (Online)</i>
Meets	<b>110-1:</b> MWF 10:00 - 10:50 (AL 303) <b>110-2:</b> MWF 2:00 - 2:50 (AL 307) <b>191-1:</b> T 8:00 - 8:50 (MB 203) <b>191-2:</b> T 9:00 - 9:50 (MB 203) <b>191-3:</b> T 10:00 - 10:50 (MB 203)
Credits	3
Prerequisites	Ma ACT 26+, SAT 640+, CpS 109, or Ma 103

## Course Description

Introduction to computer science through use of a high-level programming language as vehicle for understanding capabilities and limitations of computers. An emphasis on the object-oriented paradigm, including basic class design principles. Development of problem-solving skills through programming exercises.

## COVID-19 Policies

**Masks / Physical Distancing** - Compliance with all University COVID-19 policies is required.

**Attendance** - Attendance is required in accordance with University policy (either virtual or in person). If a student cannot attend in-person, the student must notify the professor that they plan to attend virtually either at the regularly scheduled class time or later.

**Office Hours** - Office visits will be virtual unless a student needs to meet in person. In-person office visits must be scheduled via [Calendly](#).

## Course Context

This course fulfills the following objectives of the Computer Science department:

- CS1. Design and implement efficient solutions to problems in various domains.
- CS2. Demonstrate understanding of fundamental concepts in computer science.
- CS3a. Communicate technical information effectively.

## Course Goals

The goals this semester are:

- To strengthen problem-solving abilities
- To develop debugging skills
- To develop an appreciation for the power of abstraction as a tool for managing complexity
- By learning to create small systems, better appreciate the genius of the Creator of the Universe

## Assignments

**Quizzes** - Quizzes cover topics that have been recently introduced in class to check student comprehension of the presented materials. Quizzes last approximately 10 minutes and are graded in class.

**Programs** - Programs are larger-scale individual assignments that require original thinking to solve a nontrivial problem using newly-acquired concepts and skills.

**Labs** - Lab assignments are small to medium program exercises to give you practical experience with the algorithms and theory discussed in class.

**Lab Tests** - Lab tests are given in conjunction with traditional written exams. Lab tests check your comprehension of the practical applications of the topics covered in class.

**Tests** - Tests cover all theoretical and practical topics covered in class. In general, tests are 50 questions and take approximately 50 minutes. Students should expect a variety of questions covering theory, practical application and programming questions.

## Grading

Assignment Type	#	Points	Total Points
Quizzes	12**	10	90
Labs	11*	10	100
Programs	6	60	360

Assignment Type	#	Points	Total Points
Lab Tests	3*	100	200
Tests	2	75	150
Final	1	100	100
		Total	1000

\* Lowest Grade Dropped

\*\* Lowest Three Grades Dropped

Grade	Minimum	Maximum
A	89.5	100
B	79.5	89
C	69.5	79
D	59.5	69
F	0	59

**Program Grading** Learning to program is important, but so is learning to write programs which are readable by others. For this reason programs are graded as follows:

- *60% Correctness*: Does the program produce correct results? Does it run according to specification. Attention to the written specifications is vitally important.
- *20% Style*: Is the code written according to the style guidelines and the instructors requirements? Consistency and attention to detail are important. The goal is clear, understandable and thoughtfully commented code.
- *15% Reports*: Every program and lab is accompanied by a written report showing your understanding of the deliverable you produced. Learning to communicate technical matters to others is the goal.
- *5% Presentation*: Was the program submitted according to instructions?

As this is a course designed to introduce you to programming, the experience gained from completing programming assignments is invaluable to cementing your understanding of the course material. Students are expected to submit a reasonable attempt on all programming assignments. Failure to do so may result in a penalty of up to one letter grade on the final course grade.

## Classroom Department

I realize this is college, and almost everyone is perpetually exhausted. If you find yourself falling asleep, feel free to stand up and walk to the back of the class to help yourself stay awake. If you need to go to the restroom, please exit to the rear of the class to avoid disturbing others. Please refrain from extended conversations in class so as not to disturb your fellow classmates. Asking your seatmate what is missing from your notes / example is fine; discussing lunch / non classroom topics is not.

Please be respectful of others around you while in class.

## **Deadlines / Late Work**

Departmental Late Policy: <https://cs.bju.edu/academics/policies/late-work-policy/>

The instructor reserves the right to change assignment due dates as deemed necessary. Assignments are due, electronically, by 11:59 pm of the date posted in the course schedule unless otherwise noted.

Each student is given 1 free late waiver that allows them to turn in one assignment, at their discretion, up to 1 week after the deadline with no penalty. In order to use the free late, the student must 1) notify the professor before the deadline that they intend to use the free late, 2) describe the progress that they have made towards completion of the assignment and 3) provide an approximate date of delivery. Due to grading constraints during finals week, the professor reserves the right to shorten the late period for end of semester projects.

## **Getting Help**

Students struggling with an assignment or concepts in the class are encouraged to ask the instructor for assistance either:

- in class
- before / after class
- during office hours
- via email
- via Teams (I typically make myself available via Teams to answer questions M-F from 9:00 pm to 10:30 pm)

In order to maximize your opportunity to receive help and receive the best possible grade on an assignment / in the course:

- Start assignments early. This will give you more opportunities to realize you don't fully understand a concept and ask for assistance.
- Don't wait until the night before an assignment is due to ask questions. The night an assignment is due typically sees a mad rush of questions, and I answer questions in the order that I receive them. There is no guarantee that I will be able to answer your question before the submission deadline.
- Request feedback. I cannot tell you what grade I would give to your particular solution for an assignment, but I can offer comments for how your solution can be improved.

## **Handbook Policies**

Compliance with student handbook policies is expected during class.

## **Accommodations**

Students needing accommodations due to a learning disability (visual, auditory, etc.) should provide an accommodation form obtained from the Academic Resource Center as soon as possible. Accommodations cannot be given without a form provided by the Academic Resource Center.

# Academic Honesty and Integrity Policy

Cheating on assignments and tests is forbidden. All work is to be done individually unless group work is explicitly permitted. No collaboration is allowed on tests. For regular individual assignments, we expect that the submitted work represents the student's own intellectual effort, defined as follows:

1. The program was written primarily by the student. This means that most of the code (aside from starting code provided by the instructor) must have been crafted, not copied, by the student.
2. External resources used, whether electronic or from another human, must be documented as follows:
  - Code snippets copied from online resources must be documented by a comment just above the copied snippet giving the URL of the page containing the source.
  - Explanatory help or advice regarding the design or implementation of the solution received from people other than the instructor must be documented in a report accompanying the assignment submission. This report must detail:
    - Source of information (e.g., name/email of the person who helped)
    - Relevance (i.e., how this resource helped and/or what it provided)
  - Note that students must not consult a solution to the assignment as a resource in crafting their own solution, nor share their own solution with another student. Doing so constitutes cheating.
3. The student must be able to explain, on demand, the entirety of the program on both the syntactic and semantic level.

Not all kinds of programming assignments require the same demonstration of personal intellectual effort. In the absence of any specific instructions, students should assume that at a minimum:

- For individual lab assignments, requirements 1 and 3 apply.
- For individual programming assignments, all three requirements apply.
- For group programming projects, only requirement 3 applies.

Failure to comply with any relevant integrity requirement constitutes cheating. Such incidents will be reported to the academic integrity committee. To avoid trouble:

- Do not look at another student's program code when seeking assistance. On the other hand, if another student is seeking help from you, never use your own program code as an example. The only acceptable reason another student may look at your code is to help you find a problem in your program.
- Do not write program code while another student (or lab assistant) is sitting with you. You may work out designs in pseudocode on paper with another student, but you must write program code by yourself.
- When seeking assistance from another person on a program assignment, always get his/her name so you can fulfill the documentation requirements.

## Copyright Policy

Copyright 2020/2021 Ethan McGee as to this syllabus and all lectures. Students are prohibited from selling (or being paid for taking) notes during the course to, or by any person, or commercial firm without the express written permission of the professor teaching the course.