

CpS 230 Computer Systems *Syllabus*

Instructor:	Mr. Jordan Jueckstock	Email:	jpjuecks@bju.edu
Office:	CpS Office, Mack Library	Telephone:	Ext. 2279 (Office)
Office Hours:	MWThF: 10-10:50 a.m. T: 12-1 p.m.		

Textbook: Bryant & O'Hallaron, *Computer Systems: A Programmer's Perspective*, 2nd ed. (2011).

Course Information

Description

Assembly language, interrupts, registers, memory addressing techniques, parameter passing mechanisms and the relationship between high-level languages and the computer. Both semesters, three credits. *Prerequisite:* CpS 110.

Context

This course addresses the following learning outcomes of the computer science major:

- **CS1** - Design and implement efficient solutions to problems in various domains.
- **CS2** - Demonstrate understanding of fundamental concepts in computer science; *including*:
 - language translation
 - limitations of computers
 - von Neumann architecture
 - memory hierarchy
 - quality data representation

This course further addresses the following learning outcomes of the engineering major:

- **1.3.12** - Write programs.
- **1.4.1** - Apply additional depth of knowledge in engineering topics of interest to the student.

Goals

This course will provide:

1. Understanding of computer systems architecture
2. Understanding of data representation in a computer
3. Understanding of the x86 and IBM-PC architecture
4. Skill in programming x86 processors at the assembly level
5. Appreciation for the importance of assembly language

Assignments

In-Class Quizzes: taken and graded in class; closed-book/notes unless otherwise announced. The lowest quiz grade is automatically dropped.

Tests: taken in class using machine gradable forms; closed-book/notes unless otherwise announced.

Homework: take-home written exercises that reinforce key non-programming concepts and skills.

Labs: small-scale individual programming assignments that introduce ideas that you will apply on a larger scale in the programming assignments.

Programs: larger-scale individual programming assignments that require original thinking to solve a non-trivial problem using newly-acquired concepts and skills.

Team Project: a large scale team programming assignment that requires both mastery of the course material and extensive teamwork and communication skills.

Grading

Grade reports will be available via BJUOnline. The following is a tentative breakdown of how the final grade is calculated. (*Note that exact point totals may be **subject to change**.*)

Grading:			
#:	Item:	Points:	Total:
11	In-class Quizzes (<i>lowest grade dropped</i>)	10	100
4	Take-home homework worksheets/exercises	20-40	100
10	Lab exercises (<i>programming</i>)	10-25	150
2	Programs	50	100
1	Team Project	250	250
3	Tests (<i>including the final exam</i>)	100	300
Total Points:			1010

Final grades will be computed on a strict 10% scale with **no rounding** (i.e., 900+ for an A, 800-899 for a B, 700-799 for a C, and 600-699 for a D).

However, all students will start the semester with **5 grace points** that will be added to their point totals when computing final grades. Note that these are *grace* points, not *birthright* points: the instructor may rescind a student's grace points at any time prior to finals week at his discretion (the student will be notified immediately, of course).

Policies

Deadlines and Late Work

Unless otherwise specified by the instructor, **assignments can receive full credit only if submitted in full by the beginning of class on the day due.** A 25% penalty will be applied if the assignment is not turned in on time. No credit is possible after one week. If you anticipate trouble on an assignment, see the instructor as soon as possible for assistance.

Programs may be submitted late only by approval of the instructor. The instructor will allow this only for students who formally request permission to submit the program late. The request must be made by email, and should report the number of hours invested, summarize the work completed to date, include a description of problems encountered, and include a copy of the program code. The request must be submitted by the original assignment deadline. Requests indicating little effort invested will be denied.

There will be no makeup quizzes except for University authorized absences *for which the student made prior arrangements with the instructor.* Students who are not present for a quiz will receive a 0. The lowest quiz grade will be dropped at the end of the semester.

Attendance

You are expected to be in class and come to class on time. **Attendance will be kept.** If you are late, you will be marked late. If you are more than twenty minutes late, or leave class early without making prior arrangements, you will be marked absent. *The burden of proof will be on the late student to prove that he was not more than twenty minutes late and that he was actually present.*

For planned absences, you are expected to notify the instructor via email **a week ahead of time.** Written assignments and scheduled tests should be completed **before** your planned absence; please contact the instructor to make arrangements for doing so. It is your responsibility to check in advance of an absence to verify what is due!

For absences due to incapacitating illness or emergency, you should contact the instructor as soon as you realize you will not be in class in order to make arrangements for making up any graded work without penalty. If you contact the instructor in a timely fashion, you will be able to make arrangements for making up any tests without penalty for the first occurrence. Each subsequent time a test is missed because of incapacitating illness or emergency, an additional 10 percent grade penalty for that test will be incurred.

If you are absent on a day when you have been informed in advance that work is due, then late-work policies apply regardless of the nature of the absence.

Accommodations for Students with Disabilities

If you have a documented learning disability or if you are impaired in some way (auditory, visual, cognitive, neurological, or physical), please let your instructor know this within the first week of the course so that any necessary adjustments can be made before you get behind. See the guidelines on accommodations eligibility available from [Academic Resource Center](#) for details.

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:
 - a. 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.
 - b. 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:
 - i. **Source** of information (e.g., name/email of the person who helped)
 - ii. **Relevance** (i.e., how this resource helped and/or what it provided)
 - c. 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.