UCR CS100 - Software Construction

Overview

CS 100 is a course created to teach you one simple principle, “develop twice, code once”. The purpose of this course is to teach you to plan out your development before you begin to code. Additionally, this course will teach you common unix based tools, design patterns to help you structure your code in an efficient manner, documentation, coding standards, and current industry terms and methodologies.

Catalog Description

Emphasizes development of software systems. Topics include design and implementation strategies; selection and mastery of programming languages, environment tools, and development processes. Develops skill in programming, testing, debugging, performance evaluation, component integration, maintenance, and documentation. Covers professional and ethical responsibilities and the need to stay current with technology.

Contacts

Name	Email	Position	Office Hours
Brian Crites	bcrit001@ucr.edu	Instructor	TBD
Christina Pavlopoulou	cpavl001@ucr.edu	TA Sec 21	TBD
		TA Sec 22	TBD
Patrick Le	lle018@ucr.edu	Grader	N/A

Textbook

Required: None
Suggested: Design Patterns: Elements of Reusable Object-Oriented Software

Course Access

All homework, labs, assignments, and course slides will be posted to iLearn.

Coursework & Grading (Subject to Change)

Labs	40%
Projects	30%
Exams	30%

Code Requirements:

All labs and assignments **must** be compilable on hammer.cs.ucr.edu, and should follow CalTech's coding guidelines.

Labs

Lab attendance is required, and is worth 20% of your total lab grade. Labs are to be done individually unless otherwise stated by the lab assignment or the lab TA. All labs need to be checked off by your TA either during lab hours or during your TA's office hours that week. You should come to TA office hours with your lab completed and ready to demo.

Assignments

Assignments **must** be completed in teams of two and a penalty of 10% will be deducted for every day the assignment is late, with exceptions for documented emergencies. For clarity, this means that an assignment that is 3 days late would incur a 30% penalty. It is up to students to inform the instructor when late assignments have been submitted.

Collaboration

Engineering is a team sport, because of this it is acceptable to get feedback from your peers on the correctness of your solutions and ideas. However you are not allow you to ask for or provide complete solutions to problems. Remember that doing the required work for others only makes them less likely to pass exams (and is a violation of UCR's academic integrity policy)

Tentative Schedule (Subject to Change)

Week 1	Agile	Git	
	Kanban		
	User Stories		

Week 2	Inheritance & Polymorphism	Bash	
	Inheritance & Polymorphism		
	Inheritance & Polymorphism		
Week 3	Composite Pattern	GDB	Assignment 1
	Composite Pattern		
	Composite Pattern		
Week 4	Strategy Pattern	Composite & Strategy Pattern	Assignment 2
	Strategy Pattern		
	Decorator Pattern		
Week 5	Decorator Pattern	Decorator Pattern	
	Abstract Factory Pattern		
	Abstract Factory Pattern		
Week 6	Review	Abstract Factory Pattern	Assignment 3
	Exam 1		
	Command Pattern		
Week 7	Command Pattern	Command Pattern	
	Command Pattern		
	Iterator Pattern		
Week 8	Iterator Pattern	Iterator Pattern	Assignment 4
	Iterator Pattern		
	Visitor Pattern		
Week 9	Visitor Pattern	Visitor Pattern	
	Visitor Pattern		
	Ethics		

Week 10	Review		
	ABET Exam		
	Exam 2		