Course Outline

2026-2027 Catalog

CSCI 36 - Programming Project

Catalog Description

Transfer Status: CSU
Prerequisite: CSCI 21
Unit(s): 2.00
Lecture: 17.00 Contact hours/34.00 Out of class hours/51.00 Total hours/1.00 Unit(s)
Lab: 51.00 Contact hours/0.00 Out of class hours/51.00 Total hours/1.00 Unit(s)
Total: 68.00 Contact hours/34.00 Out of class hours/102.00 Total hours/2.00 Unit(s)
Course Description: This is a capstone programming project course integrating software engineering processes, project management, and computer programming skills. Topics include problem formulation, requirements elicitation, design, tools and frameworks, implementation and testing, version control, integration, and documentation. Emphasis will be placed on teamwork to design, develop, and deliver software projects that solve specific problems.

Objectives

Upon successful completion of this course, the student should be able to:

1. Analyze and determine requirements for a software system to solve a specific problem.
2. Produce a detailed design for a software system that meets a specific set of requirements.
3. Measure and track progress of a software development project.
4. Work with a team to develop and deliver a software system that satisfies a specific set of requirements.
5. Use a version control tool to manage a software project with multiple team members and multiple revisions.
6. Write software documentation that is clear, concise, and consistent with industry standards.

Course Content

Topic Titles / Suggested Time Topic

Lecture

Topics	Lec Hrs
Software Project Planning	1.00
Building a Project Schedule	1.00
Software Project Design	2.00
Requirements Elicitation	1.00
Version Control	2.00
Unit Testing	1.00
Documentation	1.00
Software Development Methodologies	1.00
Code Reviews	1.00
Use Cases	1.00
Integration	1.00
Selecting Tools and Frameworks	1.00
Team Communications	1.00
Release Management	1.00
Measuring and Tracking Progress	1.00
Total Hours:	**17.00**

Lab

Topics	Lab Hrs
Software Project Planning	3.00
Team Communications	3.00
Building a Project Schedule	3.00
Requirements Elicitation	3.00
Use Cases	3.00
Software Project Design	6.00
Software Development Methodologies	3.00
Selecting Tools and Frameworks	3.00

Topics	Lab Hrs
Version Control	6.00
Unit Testing	3.00
Documentation	3.00
Measuring and Tracking Progress	3.00
Code Reviews	3.00
Integration	3.00
Release Management	3.00
Total Hours:	**51.00**

Methods of Instruction

A. Demonstrations
B. Guest Speakers
C. Homework: Students are required to complete two hours of outside-of-class homework for each hour of lecture
D. Lecture
E. Reading Assignments
F. Computer Programming

Methods of Evaluation

A. Exams/Tests
B. Portfolios
C. Projects

Examples of Assignments

Reading Assignments

1. Read the chapter in your text on the Extreme Programming (XP) methodology. Be prepared to discuss the concepts of test-driven development (TDD) and pair programming in class.
2. Read the article on requirements engineering provided by the instructor. Be prepared to explain in class the differences between user requirements and system requirements.

Writing Assignments

1. Select a Google programming style guide (any language of your choice) as a reference, and create a 1-2 page "cheat sheet" for a software development team to use as a style guide for a project. Focus your "cheat sheet" on the most critical components of source code, such as naming, functions, comments, and formatting.
2. Using the user and system requirements provided by the instructor, draft a brief software specification design document (2-3 pages) that includes a description of the desired application, criteria for completion, projected milestones, and core functionality.

Out-of-Class Assignments

1. Using your team's project specification, research the tools and frameworks that could be used to implement the project. Be prepared to make recommendations to your team at the next class meeting.
2. Complete your assigned code review, and record your findings in the reporting tool that your team has selected. Be prepared to discuss your code review at your next team stand-up meeting.

Recommended Materials of Instruction

Tsitoara, Mariot. (2019). Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. *APress, 1st.* 978-1484253120.
Project Management Institute. (2021). A Guide to the Project Management Body of Knowledge. *Project Management Institute, 7th.* 978-1628256642.
Rigby, Darrell; Elk, Sarah; Berez, Steve. (2020). Doing Agile Right: Transformation Without Chaos. *Harvard Business Review Press, 1st.* 978-1633698703.

Minimum Qualifications

Computer Science (Masters Required)
Computer Information Systems

Created/Revised by: Sathrum, Luke
Date: 05/02/2022