

Programming for STEM

Overview

This course expands the practice of software development in a variety of settings, so that students acquire a broad set of programming skills and a deeper understanding of software engineering principles. Students learn to plan, design and implement relatively large programming projects that require background research and team work. Topics include simulations, games and interactive on-line applications. Robust program design and sound software engineering practices are emphasized throughout the course.

Objectives

- Explain the basic structure and organization of a medium-size program.
- Design, implement and debug programs of several hundred to a few thousand lines of code.
- Make reasonable design choices for algorithms and data representation.
- Use a variety of data and control structures to make programs clear and efficient.
- Organize code hierarchically.
- Demonstrate familiarity with the functions in the standard library of one or more languages.
- Properly use documentation and on-line resources to resolve programming questions.
- Work efficiently in a team to build programs collaboratively.
- Demonstrate an understanding of integrity and ethics pertaining to computing professions

Assessment

This course is project-based. Students will need to keep a Journal detailing all the steps of the design and implementation of the projects, as well as reflections on their progress. Weekly progress assessments will be based on the Journal Entries and Work Ethic. Four 9-week projects will also be assessed on outcome, presentation and quality of the development process.

Course Essentials

Equipment	Cost/Unit
Classroom set of computers	\$0 if you already have some, \$500-600 per computer if you need to purchase
Software	All the software used in this course is open source and widely available free of charge, but installation of software in local computers is required.
Prerequisites	Intro to Computational Thinking (LSU Partnership) AND one of the following: Data Manipulation and Analysis or Interactive Computing

Outline:

Unit 1: Review of Programming Basics	Programming languages. Development tools. Program design. Program components (variables, functions, conditions, loops). Project: Multi-scene, multi-character, procedurally-generated animation.
Unit 2: Advanced programming features	Map, reduce, and higher order functions. Abstract data types. Algebraic types. Recursive data structures. Project: Extensions to the Towers of Hanoi algorithm
Unit 3: Grid-based games	Search algorithms, pathfinding, graph traversals. Collision detection. Suggested projects: Snake, Pacman, Tetris, or similar. Alternatively: virtual robot map navigation.
Unit 4: Group-project	Creating requirement proposals. Estimating required work. Planning strategies. Team roles. Testing and debugging strategies. Student chosen project (Evidence that project can be completed in the given time must be provided.)