

CS 225: OBJECT ORIENTED PROGRAMMING

Citrus College Course Outline of Record

| Heading | Value |
|-------------------------------|---------------------------------|
| Effective Term: | Fall 2024 |
| Credits: | 3 |
| Total Contact Hours: | 72 |
| Lecture Hours : | 54 |
| Lab Hours: | 18 |
| Hours Arranged: | 0 |
| Outside of Class Hours: | 108 |
| Total Student Learning Hours: | 180 |
| Prerequisite: | CS 111. |
| Strongly Recommended: | Intermediate algebra or higher. |
| Transferable to CSU: | Yes |
| Transferable to UC: | Yes - Approved |
| Grading Method: | Standard Letter |

Catalog Course Description

This course introduces the discipline of computer science using a high level language, C++, utilizing programming and practical hands-on problem solving. Topics include the use of functions and parameter passing, simple I/O, control structures, user-defined data types, arrays, searching and sorting, algorithms and debugging strategies, data abstraction, intro to pointers, concept of types and software development methods. 54 lecture hours, 18 lab hours.

Course Objectives

- Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions
- Use pseudocode or a programming language to implement, test, and debug algorithms for solving simple problems
- Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today
- Demonstrate different forms of binding, visibility, scoping, and lifetime management

Major Course Content

1. Basic syntax and semantics of a higher-level language
2. Variables, types, expressions and assignment
3. Simple I/O
4. Conditional and iterative control structures
5. Functions and parameter passing
6. Structured decomposition
7. Problem-solving strategies
8. The role of algorithms in the problem-solving process
9. Implementation strategies for algorithms
10. Debugging strategies
11. The concept and properties of algorithms

12. History of programming languages
13. Brief survey of programming paradigms
14. Procedural languages
15. The conception of types as a set of values together with a set of operations declaration models (binding, visibility, scope and lifetime)
16. Overview of type-checking
17. Arrays and Structures
18. Intro to Pointers
19. Intro to Classes

Lab Content

The student will be assigned lab work for each category below.

1. Sequence
 - a. Logical flow of the program
 - b. Flow charts
 - c. Pseudocode
2. Selection
 - a. if statement
 - b. select statement
3. Repetition
 - a. while statement
 - b. for statement
 - c. do while statement

Suggested Reading Other Than Required Textbook

The student will visit several programming online websites in order to read documentation about object oriented programming languages.

Examples of Required Writing Assignments

The student will create a flowchart and a pseudocode before implementing the programming code for any given assignment.

Examples of Outside Assignments

Students will be required to complete the following types of assignments outside of the regular class time:

- Study course concepts - Answer various programming questions - Practice skills (i.e., writing programs and creating flowcharts).
- Read required materials - Solve programming problems - Create programs that apply Object-Oriented programming techniques

Instruction Type(s)

Lab, Lecture, Online Education Lab, Online Education Lecture