CS 232: PROGRAMMING CONCEPTS AND METHODOLOGY II

Citrus College Course Outline of Record

Heading	Value
Effective Term:	Fall 2021
Credits:	3
Total Contact Hours:	72
Lecture Hours :	54
Lab Hours:	18
Hours Arranged:	0
Outside of Class Hours:	108
Prerequisite:	CS 225.
Transferable to CSU:	Yes
Transferable to UC:	Yes - Approved
Grading Method:	Standard Letter

Catalog Course Description

Application of software engineering techniques to the design and development of large programs; data abstraction and structures and associated algorithms. 54 lecture hours, 18 lab hours.

Course Objectives

- Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables.
- Implement, test, and debug simple recursive functions and procedures.
- Evaluate tradeoffs in lifetime management (reference counting vs. garbage collection).
- Explain how abstraction mechanisms support the creation of reusable software components.
- Design, implement, test, and debug simple programs in an object-oriented programming language.
- Compare and contrast object-oriented analysis and design with structured analysis and design.

Major Course Content

1. Programming Fundamentals (PF)
2. Fundamental data structures
   a. Primitive types
   b. Arrays
   c. Records
   d. Strings and string processing
   e. Data representation in memory
   f. Static, stack, and heap allocation
   g. Runtime storage management
   h. Pointers and references
   i. Linked structures
   j. Implementation strategies for stacks, queues, and hash tables
   k. Implementation strategies for trees and graphs
   l. Strategies for choosing the right data structure
3. Recursion
   a. The concept of recursion
   b. Recursive mathematical functions
   c. Simple recursive procedures
   d. Divide-and-conquer strategies
   e. Recursive backtracking
   f. Implementation of recursion
4. Programming Languages (PL)
5. Declarations and types
   a. The conception of types as a set of values together with a set of operations
   b. Declaration models (binding, visibility, scope, and lifetime)
   c. Overview of type-checking
   d. Garbage collection versus Automatic Resource Counting
6. Abstraction Mechanisms
   a. Procedures, functions, and iterators as abstraction mechanisms
   b. Parameterization mechanisms (reference vs. value)
   c. Activation records and storage management
   d. Type parameters and parameterized types - templates or generics
   e. Modules in programming languages
7. Object-oriented programming
   a. Object-oriented design
   b. Encapsulation and information-hiding
   c. Separation of behavior and implementation
   d. Classes and subclasses
   e. Inheritance (overriding, dynamic dispatch)
   f. Polymorphism (subtype polymorphism vs. inheritance)
   g. Class hierarchies
   h. Collection classes and iteration protocols
   i. Internal representations of objects and method tables
8. Software Engineering (SE)
9. Software design
   a. Fundamental design concepts and principles
   b. Design strategy

Lab Content

Students will be assigned lab work for each category below.

1. Object Based Programming
2. Object Oriented Programming
3. Polymorphism
4. Inheritance

Suggested Reading Other Than Required Textbook

www.htdp.org (i.e., How to design programs)

Examples of Required Writing Assignments

Create a written program that a school teacher can use to track student scores.

Examples of Outside Assignments

Create a GUI (i.e., graphical user interface) that allows the user to enter two numbers and then determines the minimum between the two simultaneously.

Instruction Type(s)

Lecture, Lab, Online Education Lecture, Online Education Lab