GAVILAN COLLEGE

5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 46 DIVISION: 50 ALSO LISTED AS:

TERM EFFECTIVE: Fall 2019 CURRICULUM APPROVAL DATE: 04/09/2019

SHORT TITLE: C++ PROGRAMMING II

LONG TITLE: C++ Programming II

Units	Number of Weeks		Contact Hours/Week		Total Contact Hours
3	18	Lecture:	2	Lecture:	36
		Lab:	3	Lab:	54
		Other:	0	Other:	0
		Total:	5	Total:	90

COURSE DESCRIPTION:

This course is a continuation of CSIS 45, intended for students majoring in programming and/or planning to transfer to a 4-year college or university Computer Science program. The course will cover topics discussed in CSIS 45 in more detail. In addition the course will cover more advanced C techniques such as pointers, recursion, and linked lists. Special emphasis will be placed on C++ features such as classes, objects, templates and operator overloading. This course has the option of a letter grade or pass/no pass. (C-ID: COMP 132) PREREQUISITE: CSIS 45 or CSIS 5 or CSIS 24 or equivalent.

PREREQUISITES:

Completion of CSIS 5, as UG, with a grade of C or better.
OR
Completion of CSIS 45, as UG, with a grade of C or better.
OR
Completion of CSIS 24, as UG, with a grade of C or better.

COREQUISITES:

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES

L - Standard Letter Grade
P - Pass/No Pass

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:

- 02 - Lecture and/or discussion
- 03 - Lecture/Laboratory
- 04 - Laboratory/Studio/Activity
- 04B - Laboratory - LEH 0.75
- 05 - Hybrid
- 72 - Dist. Ed Internet Delayed
- 73 - Dist. Ed Internet Delayed LAB
- 73B - Dist. Ed Internet LAB-LEH 0.75

STUDENT LEARNING OUTCOMES:

1. Discuss the representation and use of primitive data types and built-in data structures.
Measure of assessment: exams, discussion, homework exercises
Year assessed, or planned year of assessment: 2018
Semester: Fall

2. Describe and demonstrate how the various data structures are allocated and used in memory.
Measure of assessment: exams, programming problems, homework exercises
Year assessed, or planned year of assessment: 2018
Semester: Fall

3. Describe and utilize common applications for a variety of data structures.
Measure of assessment: exams, programming problems, homework exercises

CONTENT, STUDENT PERFORMANCE OBJECTIVES, OUT-OF-CLASS ASSIGNMENTS

Curriculum Approval Date: 04/09/2019

Lecture content 36 Hours.

4 Hours

Lecture Content: Data Abstraction: The Walls, Recursion: The Mirrors, and Recursion as a Problem-Solving Technique.

Student Performance Objectives: Define an abstract data type (ADT) and explain the difference between an ADT and a data structure. Explain the steps you should take when designing an ADT. Utilize the UML to describe classes at various stages of their development. Discuss recursion and utilize recursive solutions in solving programming exercises. Define the concept of formal grammars and explain its use. Discuss the problem-solving technique of backtracking.

4 Hours

Lecture Content: Array-Based Implementations and Link-Based Implementations.

Student Performance Objectives: Define 'stubs' and explain their benefit. Discuss what happens to the array 'items' when the method 'add' cannot add another entry to it, because it is already full. List an advantage and a disadvantage of calling the method 'getFrequencyOf' from 'contains'. Define the following and state the relationship between each: a 'node', the 'head pointer', and a 'linked chain'.

4 Hours

Lecture Content: Stacks and Stack Implementations.

Student Performance Objectives: Discuss the benefits of a stacks last-in, first-out behavior. Explain the significance of the precedence tests in the infix-to-postfix conversion algorithm. Discuss the relationship between stacks and recursion. Describe the changes to a stack implementation that are necessary to replace a fixed-stack array with a resizable array. Discuss the advantages and disadvantages of an array-based implementation of the ADT stack as compared to a link-based implementation.

4 Hours

Lecture Content: Lists and List Implementations.

Student Performance Objectives: Write specifications for a list whose operations 'insert', 'remove', 'getEntry', and 'setEntry' always act at the end of the list. Discuss the advantages and disadvantages of an array-based implementation of the ADT list as compared to a link-based implementation.

4 Hours

Lecture Content: Midterm. Algorithm Efficiency and Sorting Algorithms and Their Efficiency.

Student Performance Objectives: State the difficulties with comparing programs instead of algorithms. Explain how you would assess an algorithms efficiency. Describe the functions of a variety of sorting processes; such as the selection sort, the bubble sort, the insertion sort, the quick sort, the merge sort, and the radix sort.

4 Hours

Lecture Content: Sorting Lists and Their Implementations, Queues and Priority Queues, and Queue Implementations.

Student Performance Objectives: Explain the purpose of the ADT sorted list and itemize its operations. Summarize the efficiencies of the operations for array-based and link-based implementations of the ADTs list and sorted list. Discuss the ADT queue operations as well as the ADT priority queue operations. Explain the function of a position-oriented ADT vs. a value-oriented ADT.

4 Hours

Lecture Content: Trees, Tree Implementations, and Heaps.

Student Performance Objectives: Summarize tree terminology. Discuss the operations that define the ADT binary tree and binary search tree. Define a 'heap' and list its operations.

4 Hours

Lecture Content: Dictionaries and Their Implementation and Balanced Search Trees.

Student Performance Objectives: Discuss the pros and cons of the linear implementations (array-based and link-based) and a nonlinear link-based (binary search tree) implementation of the ADT dictionary. List various ADT dictionary operations. Define 'hashing' and explain what a hash function does. State the advantages of implementing the ADT dictionary with a 2-3 tree instead of a binary search tree. Discuss the reasons why it is desirable in many applications to use one of several variations of the basic binary search tree vs. a minimum-height binary search tree. Explain why a node in a red-black tree requires less memory than a node in a 2-3-4 tree.

2 Hours

Lecture Content: Graphs and Processing Data in External Storage.

Student Performance Objectives: List ADT graph operations. Explain the two most common implementations of a graph. Discuss various useful applications of graphs. Discuss how to sort the data in an external file by modifying the merge sort algorithm. Discuss how to search an external file by using generalizations of the hashing and search-tree schemes.

2 Hours

Final.

Lab Content 54 Hours

Lab Content: Dictionaries and Their Implementation and Balanced Search Trees.

Student Performance Objectives: Complete exercises and programming problems related to Dictionaries and Their Implementation and Balanced Search Trees. Apply more than one hash function to a single search key.

3 Hours

Lab Content: Graphs and Processing Data in External Storage.

Student Performance Objectives: Complete exercises and programming problems related to Graphs and Processing Data in External Storage. Demonstrate the two most common implementations of a graph. Apply various useful applications of graphs. Demonstrate how to sort the data in an external file by modifying the merge sort algorithm. Demonstrate how to search an external file by using generalizations of the hashing and search-tree schemes.

METHODS OF INSTRUCTION:

Lecture, computer demonstration, hands-on exercises and practices.

OUT OF CLASS ASSIGNMENTS:

Required Outside Hours: 24

Assignment Description: Reading textbook.

Required Outside Hours: 48

Assignment Description: Homework: Complete exercises and programming problems. Work on programming projects.

METHODS OF EVALUATION:

Problem-solving assignments

Percent of total grade: 40.00 %

Problem-solving demonstrations: 30% - 60% Homework problems, Programming projects, Quizzes, Exams

Skill demonstrations

Percent of total grade: 50.00 %

Skill demonstrations: 40% - 60% Demonstration, Performance exams

Objective examinations

Percent of total grade: 10.00 %

REPRESENTATIVE TEXTBOOKS:

Frank M. Carrano and Timothy M. Henry. Data Abstraction & Problem Solving with C++ (7th Edition). Pearson,2017.

Reading Level of Text, Grade: 12 Verified by: MS Word

ARTICULATION and CERTIFICATE INFORMATION

Associate Degree:

CSU GE:

CSU B3, effective 201970

CSU B8, effective 201970

IGETC:

CSU TRANSFER:

Transferable CSU, effective 200630

UC TRANSFER:

Transferable UC, effective 201970

SUPPLEMENTAL DATA:
Basic Skills: N
Classification: Y
Noncredit Category: Y
Cooperative Education:
Program Status: 1 Program Applicable
Special Class Status: N
CAN:
CAN Sequence:
CSU Crosswalk Course Department: CST
CSU Crosswalk Course Number: 238
Prior to College Level: Y
Non Credit Enhanced Funding: N
Funding Agency Code: Y
In-Service: N
Occupational Course: B
Maximum Hours:
Minimum Hours:
Course Control Number: CCC000564666
Sports/Physical Education Course: N
Taxonomy of Program: 070710