GAVILAN COLLEGE

5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 27 DIVISION: 50 ALSO LISTED AS:

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

SHORT TITLE: JAVA PROGRAMMING II

LONG TITLE: Java Programming II

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

COURSE DESCRIPTION:

This course is a continuation of Java Programming I, intended for students majoring in programming and/or planning to transfer to a 4-year college or university. This course will cover topics discussed in Java Programming I in more detail. Emphasis will be placed on implementation and analysis of algorithms and abstract data types: lists, queues, stacks, arrays, trees, priority queues, heaps, tables, hashing, balanced trees, graphs, searching and sorting, and recursion. (C-ID: COMP 132) PREREQUISITE: CSIS 24 Java Programming I, or CSIS 45, or equivalent experience.

PREREQUISITES:

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

COREQUISITES:

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES

L - Standard Letter Grade

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:

02 - Lecture and/or discussion
05 - Hybrid
72 - Dist. Ed Internet Delayed

STUDENT LEARNING OUTCOMES:

1. Explain 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

Year assessed, or planned year of assessment: 2018

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

Curriculum Approval Date: 04/09/2019

3 Hours

Content: Java Review

Program Design: Pseudocode, Coding, Documentation and Style, Testing and Debugging

Object-Oriented Design: Goals, Principles, and Patterns, Object-Oriented Design Goals, Object-Oriented Design Principles, Design Patterns, Inheritance, Polymorphism and Dynamic Dispatch, Inheritance Hierarchies, Interfaces and Abstract Classes

Student Performance Objectives: Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism. Design, implement, test, and debug simple programs in an object-oriented programming language.

3 Hours

Content: Fundamental Data Structures: Using Arrays, Sorting an Array, javautil Methods for Arrays and Random Numbers, Simple Cryptography with Character Arrays, Two-Dimensional Arrays, Singly Linked Lists, Circularly Linked Lists

Student Performance Objectives:

3 Hours

Content: Doubly Linked Lists: Implementing a Doubly Linked List Class, Equivalence Testing with Arrays, Equivalence Testing with Linked Lists, Cloning Data Structures, Cloning Arrays, Cloning Linked Lists

Student Performance Objectives:

3 Hours

Content: Recursion: Illustrative Examples, The Factorial Function, Drawing an English Ruler, Binary Search

Student Performance Objectives: Describe the concept of recursion and give examples of its use. Determine when a recursive solution is appropriate for a problem.

3 Hours

Content: File Systems: Analyzing Recursive Algorithms, Further Examples of Recursion, Linear Recursion, Binary Recursion, Multiple Recursion, Designing Recursive Algorithms, Maximum Recursive Depth in Java, Eliminating Tail Recursion

Student Performance Objectives:

6 Hours

Content: Stacks, Queues, and Deques

Stacks: The Stack Abstract Data Type, A Simple Array-Based Stack Implementation, Implementing a Stack with a Singly Linked List, Reversing an Array Using a Stack, Matching Parentheses and HTML Tags

Queues: The Queue Abstract Data Type, Array-Based Queue Implementation, Implementing a Queue with a Singly Linked List, A Circular Queue, Double-Ended Queues

Deques: The Deque Abstract Data Type, Implementing a Deque, Deques in the Java Collections Framework

Student Performance Objectives:

6 Hours

Content: List and Iterator ADTs: The List ADT, Array Lists, Dynamic Arrays, Java's StringBuilder class

Positional Lists: Positions, The Positional List Abstract Data Type, Doubly Linked List Implementation, Iterators, The Iterable Interface and Java's For-Each Loop, Implementing Iterators, The Java Collections Framework

Student Performance Objectives: Describe how iterators access the elements of a container.

3 Hours

Content: Trees: General Trees, Tree Definitions and Properties, The Tree Abstract Data Type, Computing Depth and Height, Binary Trees, The Binary Tree Abstract Data Type, Properties of Binary Trees

Implementing Trees: Linked Structure for Binary Trees, Array-Based Representation of a Binary Tree, Linked Structure for General Trees, Tree Traversal, Algorithms, Preorder and Postorder Traversals of General Trees, Breadth-First Tree Traversal, Inorder Traversal of a Binary Tree, Implementing Tree Traversals in Java

Student Performance Objectives:

3 Hours

Content: Priority Queues: The Priority Queue Abstract Data Type, Implementing a Priority Queue, The Abstract Priority Queue Base Class, Implementing a Priority Queue with an Unsorted List, Implementing a Priority Queue with a Sorted List

Student Performance Objectives:

3 Hours

Content: Hash Tables: Hash Functions, Collision-Handling Schemes, Load Factors, Rehashing, and Efficiency, Java Hash Table Implementation

Student Performance Objectives: Write programs that use hash tables.

3 Hours

Content: Search Trees: Binary Search Trees, Searching Within a Binary Search Tree, Insertions and Deletions, Java Implementation, Performance of a Binary Search Tree, Balanced Search Trees, Java Framework for Balancing Search Trees, Red-Black Trees

Student Performance Objectives:

6 Hours

Content: Sorting and Selection: Merge-Sort, Divide-and-Conquer, Quick-Sort, Randomized Quick-Sort, Additional Optimizations for Quick-Sort, Studying Sorting through an Algorithmic Lens, Lower Bound for Sorting, Linear-Time Sorting: Bucket-Sort and Radix-Sort

Student Performance Objectives:

6 Hours

Content: Memory Management: Stacks in the Java Virtual Machine, Allocating Space in the Memory Heap, Garbage Collection, Memory Hierarchies and Caching

Student Performance Objectives:

2 Hours

METHODS OF INSTRUCTION:

Lecture, computer demonstration, hands on exercises and practices.

OUT OF CLASS ASSIGNMENTS:

Required Outside Hours: 34

Assignment Description: Reading the textbook.

Required Outside Hours: 70

Assignment Description: Working on sample programs, homework programs, and 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 Structures and Abstractions with Java (5th Edition). Pearson,2018.

ISBN: 0134831691

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

CSU E1, effective 201970

IGETC:

CSU TRANSFER:

Transferable CSU, effective 201570

UC TRANSFER:

Not Transferable

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: E

Maximum Hours: 3

Minimum Hours: 3

Course Control Number: CCC000562532

Sports/Physical Education Course: N

Taxonomy of Program: 070600