

CS A272: JAVA PROGRAMMING 2

Item	Value
Curriculum Committee Approval Date	11/28/2018
Top Code	070710 - Computer Programming
Units	4 Total Units
Hours	90 Total Hours (Lecture Hours 63; Lab Hours 27)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Open Entry/Open Exit	No
Grading Policy	Standard Letter (S), • Pass/No Pass (B)

Course Description

A second course in Java programming language. Topics include object-oriented design, inheritance, interfaces, abstract and anonymous inner classes, I/O & exceptions, generics, regular expressions, databases, XML, GUI construction, graphics and multimedia, Java collections, data structures, lambda expressions and multithreading. PREREQUISITE: CS A170. Transfer Credit: CSU; UC.

Course Level Student Learning Outcome(s)

1. Write programs that make appropriate use of each of the Java Collections classes and that correctly use Java's generics facility.
2. Demonstrate an understanding of recursive methods and write recursive functions to solve different programming problems.
3. Write programs that implement linked lists, array lists, stacks and queues. Apply those data structures to solving different kinds of programming problems.

Course Objectives

- 1. Design, implement, test and debug simple programs using classes and objects.
- 2. Implement, test and debug simple recursive functions and procedures.
- 3. Understand and utilize the basic Java I/O libraries.
- 4. Write programs that make appropriate use of each of the Java Collections classes.
- 5. Write programs that use arrays, records, strings, linked lists, stacks, queues and hash tables.
- 6. Use object-oriented analysis and design to apply inheritance, interfaces and composition where appropriate.
- 7. Apply different abstraction methods, such as Java generics to support the creation of reusable software components.
- 8. Design an abstract class and an appropriate set of instantiable child classes to solve a software requirement.

- 9. Design and use error handling classes employing try, catch, throw and finally.
- 10. Create an application with a Graphical User Interface

Lecture Content

Topic 1 - Fundamental Data Structures (CS 170 Review) Primitive types (int, double, char, boolean) Arithmetic, input and output and sequence Strings and string processing Selection, the if statement and the boolean type Iteration and loops Arrays and ArrayLists Topic 2 - Object-oriented programming basics Calling Methods Constructing Objects Accessor and Mutator Methods Object References, the stack and the heap Instance Variables and Encapsulation Abstraction - interface and implementation Testing the class (unit tests) The this Reference Topic 3 - Object-oriented design basics Discovering Classes Designing Good Methods Problem Solving: Patterns for Object Data Static Variables and Methods Modules and Packages Unit Test Frameworks Classes and Their Responsibilities Relationships Between Classes Topic 4 - Inheritance Inheritance Hierarchies Implementing Subclasses Overriding Methods Polymorphism Object: The Cosmic Superclass Topic 5 - Interfaces and Abstract Classes Using Interfaces for Algorithm Reuse Working with Interface Types The Comparable Interface Using Interfaces for Callbacks Inner Classes Mock Objects Event Handling Building Applications with Buttons Processing Timer Events Mouse Events Topic 6 - Input/Output and Exception Handling Reading and Writing Text Files Text Input and Output Command-Line Arguments Exception Handling Handling Input Errors Topic 7 - Recursion Triangle N umbers Recursive Helper Methods The Efficiency of Recursion Permutations Mutual Recursion Backtracking Topic 8 - Sorting and Searching Selection Sort Profiling the Selection Sort Algorithm Analyzing the Performance of the Selection Sort Merge Sort Analyzing the Merge Sort Algorithm Searching Estimating the Running Time of an Algorithm Sorting and Searching in the Java Library Topic 9 - The Java Collections Framework Linked Lists SetsMaps Stacks, Queues, and Priority Queues Stack and Queue Applications Topic 10 - Basic Data Structures Implementing Linked Lists Implementing Array Lists Implementing Stacks and Queues Implementing a Hash Table Topic 11 - Tree Structures Binary Trees Binary Search Trees Tree Traversal Red-Black Trees Heaps The Heapsort Algorithm Topic 12 - Generic Classes Generic Classes and Type Parameters Implementing Generic Types Generic Methods Constraining Type Parameters Type Erasure Topic 13 - Stream Processing and Functional Programming The Stream Concept Producing Streams Collecting Results Transforming Streams Lambda Expressions Common Algorithms Revisited Topic 14 - Practical Applications Graphical User Interfaces Advanced Input/Output Multithreading Internet Networking Relational Databases Web Applications

Lab Content

Lab 1 The Eclipse IDE Lab 2 The Command Line Lab 3 I/O Redirection and Character Encodings Lab 4 Input/Output and Exception Handling Lab 5 Recursion Lab 6 Unit Testing and Shell Commands Lab 7 Version Control Lab 8 Big-Oh Lab 9 Pseudocode Bootcamp Lab 10 Using Collections bucket.io/cs46b/lab8/index.html Lab 11 Debugging Linked Structures Lab 12 Scripts Lab 13 Trees

Method(s) of Instruction

- Lecture (02)
- DE Online Lecture (02X)

- Lab (04)
- DE Online Lab (04X)

Instructional Techniques

Lecture, demonstrations, and guided, hands-on exercises

Reading Assignments

Students will spend a minimum of 3 hours per week reading the textbook and/or other reading material assigned. Students will be expected to follow along with the exercises in the reading material.

Writing Assignments

Out of class assignments are all writing assignments.

Out-of-class Assignments

Students will spend a minimum of 5 hours per week completing weekly programming assignments.

Demonstration of Critical Thinking

Written quizzes and examinations; weekly programming exercises

Required Writing, Problem Solving, Skills Demonstration

Weekly homework assignments will be evaluated.

Eligible Disciplines

Computer science: Master's degree in computer science or computer engineering OR bachelor's degree in either of the above AND master's degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelor's degree in engineering AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in mathematics AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in any of the above AND a master's degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Master's degree required.

Textbooks Resources

1. Required Horstmann, C.. Big Java, 7th ed. Hoboken, NJ: Wiley, 2018

Other Resources

1. The Java Tutorials online at <https://docs.oracle.com/javase/tutorial/>