CSCI 0012 - PROGRAMMING CONCEPTS AND METHODOLOGY I

Catalog Description

Prerequisite: Completion of CSCI 10 with grade of "C" or better
Hours: 72 (54 lecture, 18 laboratory)
Description: Introduces the discipline of computer science using a high level language utilizing programming and practical hands-on problem solving. (C-ID COMP 122) (CSU, UC)

Course Student Learning Outcomes

- CSLO #1: Demonstrate basic proficiency in the mechanics of the Java programming language.
- CSLO #2: Apply language fundamentals in support of utilizing and creating Object-Oriented (O-O) software elements.
- CSLO #3: Originate new Object-Oriented (O-O) software, using language fundamentals and O-O fundamentals, in accordance with sound software development principles.
- CSLO #4: Apply language fundamentals, O-O fundamentals, and sound program organization techniques, for the purpose of applying them to various realistic, real-world problem solving scenarios.

Effective Term

Fall 2020

Course Type

Credit - Degree-applicable

Contact Hours

72

Outside of Class Hours

90

Total Student Learning Hours

162

Course Objectives

Lecture Objectives:
1. Analyze and explain the behavior of simple programs involving the fundamental programming constructs.
2. Choose appropriate conditional and iteration constructs for a given programming task.
3. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
4. Describe the mechanics of parameter passing.
5. Discuss the importance of algorithms in the problem-solving process.
6. Identify the necessary properties of good algorithms.
7. Create algorithms for solving simple problems.
8. Describe strategies that are useful in debugging.
9. Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.
10. Identify at least one distinguishing characteristic for each of the programming paradigms.
11. Explain the value of declaration models, especially with respect to programming-in-the-large.
12. Identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size.
13. Discuss type incompatibility.
14. Demonstrate different forms of binding, visibility, scoping, and lifetime management.
15. Defend the importance of types and type-checking in providing abstraction and safety.
Laboratory Objectives:
1. Modify and expand short programs that use standard conditional and iterative control structures and functions.
2. 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.
3. Use a programming language to implement, test, and debug algorithms for solving simple problems.

General Education Information

- Approved College Associate Degree GE Applicability
- CSU GE Applicability (Recommended-requires CSU approval)
- Cal-GETC Applicability (Recommended - Requires External Approval)
- IGETC Applicability (Recommended-requires CSU/UC approval)

Articulation Information

- CSU Transferable
- UC Transferable

Methods of Evaluation

- Objective Examinations
  - Example: Examination based on solutions to common business software problems, and evaluated for syntax and logic errors, readable, efficient, and maintainable code style, and applying Java Object Oriented Programming concepts (encapsulation, instantiation, inheritance, and polymorphism). Example: What is the difference between static and non static methods?
- Problem Solving Examinations
  - Example: Example: Write a computer program that plays rock, paper, scissor game. Rubric Grading.
- Projects
  - Example: Applying data file handling techniques for processing business data accurately and efficiently using file creation, updating, deleting, and maintenance. Example: Write an object oriented program to play the Tic=Tac-Toe game based on the given requirements. Rubric Grading.

Repeatable

No

Methods of Instruction

- Laboratory
- Lecture/Discussion
- Distance Learning

Lab:

1. Once the material on creating user defined data types has been presented, assess whether the students have learned the material by having them write a simple class definition using the following specifications: (Laboratory Objective 2) The class should be named "Person" and be public. The class has four variables: name (String), age (int), height (int), and eyeColor (String). Write at least two of the getters and two of the setters. Write a public method called "birthday" that increments the age by
2. Write a public method called "toString" that returns a String containing all the attribute of the Person in a human-readable format. Project the "interesting" solutions on the board so that other students may comment, criticize, and/or praise.

Lecture:

1. Assume the students have come to class having read the chapter on defining a class. Give a short lecture, supplemented with writing on the board or a PowerPoint presentation with handouts, showing the syntax of a Java class. Point out the major sections: class header, class variables, and methods. Discuss the differences between "public" and "private" methods. Discuss conventional "getter" and "setter" methods. Ask the students to answer the questions in the handout related to the lecture. (Objective 1)

Distance Learning

1. Through Distance Learning, the instructor will present a video lecture on writing classes. After the student views the lecture, they will then be asked to write a class called Circle with the radius as its attribute. The student will be asked to write a method for the class to calculate the area of the circle. (Laboratory Objective 2)

Typical Out of Class Assignments

Reading Assignments

1. Read & study the examples called Switch1.java and Switch2.java in "Java Programming Examples" and contrast the two versions of the switch statement and be prepared to discuss in class. 2. Read chapter on Classes to contrast classes versus objects. Review the concepts of encapsulation and instantiation. Study the examples of Java code illustrating data-hiding, member data, and methods. Be prepared to discuss in class.

Writing, Problem Solving or Performance

Laboratory assignment per week solving Business/Industry problems with Java Programming source code. Example 1: Write a program for a money exchange business, prompting for amount of U.S. dollars, asking for a country/currency (e.g. Japanese Yen, European Euro, Mexican Peso, etc.), input a currency exchange rate, calculate the equivalent amount in the new currency, validate input data for accuracy, and display the data on the screen. Apply repetition to loop through many currency conversions. Example 2: Sierra Fencing Company installs new fencing and repairs or replaces old fencing. Write a program that prompts for new or repair fencing, prompts for the cost-per-foot, prompts for fencing length, estimates the total cost, and displays the data on the screen and/or file.

Other (Term projects, research papers, portfolios, etc.)

Required Materials

- Building Java Programs: A Back to Basics Approach
    - Author: Stuart Reges, Marty Steep
    - Publisher: Pearson
    - Publication Date: 2019
    - Text Edition: 5th
    - Classic Textbook?: No
    - OER Link:
    - OER:
- Starting Out with Java
    - Author: Tony Gaddis
    - Publisher: Scott Jones
    - Publication Date: 2019
    - Text Edition: 7th
    - Classic Textbook?: No
    - OER Link:
    - OER:
- Java - An Introduction to Problem Solving and Programming
    - Author: Walter Savitch
    - Publisher: Pearson/Prentice Hall
    - Publication Date: 2017
    - Text Edition: 8th
    - Classic Textbook?: No
    - OER Link:
    - OER:

Other materials and-or supplies required of students that contribute to the cost of the course.