

### Course Outline

**COURSE:** CSIS 24                      **DIVISION:** 50                      **ALSO LISTED AS:**

**TERM EFFECTIVE:** Spring 2021                      **CURRICULUM APPROVAL DATE:** 12/8/2020

**SHORT TITLE:** JAVA PROGRAMMING I

**LONG TITLE:** Java Programming I

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

**COURSE DESCRIPTION:**

Introduction to Java programming. Includes programming fundamentals, program design, and core computer concepts. Covers the basics of object-oriented programming in the Java environment. (C-ID: COMP 122) ADVISORY: CSIS 42 or comparable programming experience.

**PREREQUISITES:**

**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
- 05 - Hybrid
- 71 - Dist. Ed Internet Simultaneous
- 72 - Dist. Ed Internet Delayed

## **STUDENT LEARNING OUTCOMES:**

By the end of this course, a student should:

1. Analyze and explain the behavior of simple programs involving the fundamental Java programming constructs.
2. Modify and expand short programs that use standard conditional and iterative control structures and functions.
3. 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 and objects.

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

Curriculum Approval Date: 12/8/2020

6 Hours

An Overview of Computers and Programming Languages

History

Programming Paradigms

Procedural Languages

Object-oriented Languages

Features of Java

Student Performance Objectives: Compare and contrast programming paradigms. Summarize the history of programming languages and describe the issues each programming language sought to address.

7 Hours

Basic Elements of Java

Variables, types, expressions and assignment

Simple I/O

Introduction to objects

Student Performance Objectives: Describe the basic elements of Java. Complete assigned exercises and programming problems.

9 Hours

Introducing data types and operators

Declarations and types

Why data types are important

Java's primitive types

Type-checking

Binding, visibility, scope, lifetime

Arithmetic operators

Relational operators

Assignment operator

Type conversion in assignment

Casting incompatible types

Student Performance Objectives: Describe programming declaration models. Define properties of a variable such as address, value, scope, persistence, size. Explain type incompatibility and the importance of type-checking.

Write simple programs using arithmetic, relational, and assignment operators.

6 Hours

Introduction to algorithms

Problem solving

Introduction to debugging

Student Performance Objectives: Create and implement algorithms for simple problems. Describe how to implement algorithms. Discuss the importance of good algorithms. Describe debugging strategies.

6 Hours

Selection Control Structures

The if statement

Nested ifs

If then else

Switch statement

Student Performance Objectives: Write simple programs using if statements, if then else statements, switch statements. Choose an appropriate selection control structure for a programming task.

6 Hours

Repetition Control Structures

For statements

Do while statements

Switch multiple-selection statement

Break and continue statements

Logical operators

Student Performance Objectives: Write simple programs using repetition control statements. Choose an appropriate repetition control structure for a programming task.

12 Hours

Functions and Parameter passing

Structured Decomposition

User Defined Functions

User-Defined Classes and ADT

Parameter passing

Introduction to Classes, Objects Methods and Strings

Student Performance Objectives: Apply the techniques of structured decomposition to break a program into smaller pieces. Describe the mechanics of parameter passing. Compare the object-oriented approach and the procedural approach to structured decomposition.

2 Hours

Final Exam

### **METHODS OF INSTRUCTION:**

Lecture, discussion, guided practice

### **OUT OF CLASS ASSIGNMENTS:**

Required Outside Hours: 36

Assignment Description:

Read assigned textbook chapter(s).

Required Outside Hours: 18

Assignment Description:

Study for exams.

Required Outside Hours: 54

Assignment Description:

Homework: Do assigned exercises and programming problems.

**METHODS OF EVALUATION:**

Writing assignments

Percent of total grade: 5.00 %

Writing assignments: 5% - 20% Written homework

Problem-solving assignments

Percent of total grade: 60.00 %

Problem-solving demonstrations: 40% - 70% Homework Problems, Exams

Skill demonstrations

Percent of total grade: 20.00 %

Skill demonstrations: 20% - 50% Class Performance, Performance Exams

Objective examinations

Percent of total grade: 15.00 %

Objective examinations: 5% - 20% Multiple Choice, True/False, Matching Items, Completion

**REPRESENTATIVE TEXTBOOKS:**

Deitel & Deitel. Java How to Program. Pearson Education, Inc.,2018.

Reading Level of Text, Grade: 12+ Verified by: E. Venable

**ARTICULATION and CERTIFICATE INFORMATION**

Associate Degree:

CSU GE:

IGETC:

CSU TRANSFER:

Transferable CSU, effective 200630

UC TRANSFER:

Transferable UC, effective 200630

**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: COMP

CSU Crosswalk Course Number: 122

Prior to College Level: Y

Non Credit Enhanced Funding: N

Funding Agency Code: Y

In-Service: N

Occupational Course: C

Maximum Hours:

Minimum Hours:

Course Control Number: CCC000564664

Sports/Physical Education Course: N

Taxonomy of Program: 070710