This course will cover the fundamentals of software development using the Java Language. The discussions of software development process will include: designing, writing source code, compiling, linking, executing, and debugging. Data types, arithmetic and logical expressions, debugging, looping, branching, modularization, dynamic memory allocation, classes and objects will be discussed in lectures and practiced through lab projects. Both console and GUI-based (Graphical User Interface) applications will be designed and created. ADVISORY: Computer Science G102, Math G030

C-ID COMP 122

JUSTIFICATION FOR COURSE:

PREREQUISITES:

COREQUISITES:

ADVISORIES:
- CS G102: Computer Software Development, Introduction
- MATH G030: Intermediate Algebra

ASSIGNED DISCIPLINES:
- Computer science

MATERIAL FEE: Yes [ ] No [X] Amount: $0.00

CREDIT STATUS: Noncredit [ ] Credit - Degree Applicable [X] Credit - Not Degree Applicable [ ]

GRADING POLICY: Pass/No Pass [X] Standard Letter [X] Not Graded [ ] Satisfactory Progress [ ]

OPEN ENTRY/OPEN EXIT: Yes [ ] No [X]

TRANSFER STATUS: CSU Transferable[ ] UC/CSU Transferable[X] Not Transferable[ ]

BASIC SKILLS STATUS: Yes [ ] No [X] LEVELS BELOW TRANSFER: Not Applicable

CALIFORNIA CLASSIFICATION CODES: Y - Not Applicable

NON CREDIT COURSE CATEGORY: Y - Not applicable, Credit Course

OCCUPATIONAL (SAM) CODE: C

REPEATABLE ACCORDING TO STATE GUIDELINES: No [X] Yes [ ] NUMBER REPEATS:

REQUIRED FOR DEGREE OR CERTIFICATE: No [ ] Yes [X]
- Associate of Arts: Liberal Arts: Emphasis in Business and Technology(Associate in Arts)
- Computer Science(Associate in Science for Transfer)
- Geography(Associate in Arts for Transfer)

GE AND TRANSFER REQUIREMENTS MET:
- CSU Transfer Course
CS G153-Java Programming, Introduction

A. Transfers to CSU
   Transfer Model Curriculum AS-T

UC Transfer Course
A. Transfers to UC

COURSE LEVEL STUDENT LEARNING OUTCOME(S) Supported by this course:

1. Code all the necessary expressions, branches, loops, functions, classes.
2. Given a set of requirements for a small business or scientific problem, prepare the software development specification.
3. Design the software components and draw flow-charts for the complex code sections.
4. 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.
5. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
6. Describe and utilize the mechanics of parameter passing.
7. Design a simple user interface to satisfy the user interactions.
8. Add the appropriate error handling routines.

COURSE OBJECTIVES:
1. Assess, analyze, and design software solutions for moderately complex business and scientific problems
2. Properly document the solution
3. Write the software code, mathematical formulas/expressions, and algorithms in the Java language
4. Eliminate coding and logic errors using sophisticated debugging tools
5. Understand and apply basic optimization techniques
6. Break a large software solution into manageable modules
7. Manage static and dynamic memory allocations
8. Interact with input, output devices and files
9. Implement applications using Object Oriented Programming (OOP) paradigms through the use of classes and objects
10. Summarize the evolution of programming languages illustrating how this history has led to the paradigms.

COURSE CONTENT:

LEcTURE CONTENT:

A. Software Development Process
   1. Collecting Requirements
   2. Preparing Specifications
   3. Designing Solution Options Including Structure Charts and Flow Charts
   4. Coding
   5. Documenting Requirements, Specifications, Solution Options, and User Guides
   6. Compiling & Linking
   7. Executing and Debugging

B. Programming Concepts
   1. Data Types
   2. Variables and Constants
3. Arithmetic and Logical Expressions  
4. Conversion of Business and Scientific Formulas to Java Mathematical Expressions  
5. Branching and Looping  
6. Modularization  
7. Functions  
8. Parameter Passing  
9. Dynamic Memory Allocation  
10. Pointers  
11. Arrays and Strings  
12. Files and Streams  
13. Object Oriented Programming (OOP) paradigm  
14. Classes and Objects  
15. Inheritance  
16. Errors and Exceptions  
17. Console Programming  
18. Event-Driven and GUI Programming  

LABORATORY CONTENT:  
1) Given a set of requirements for a small business or scientific problem, prepare the software development specification  
2) Design the software components and draw flow-charts for the complex code sections  
3) Create a project with the correct file structure  
4) Break the programs into appropriate classes  
5) Design a simple user interface to satisfy the user interactions  
6) Code all the necessary expressions, branches, loops, functions, classes  
7) Add the appropriate error handling routines  

METHODS OF INSTRUCTION:  
A. Lecture:  
B. Lab:  
C. Online:  
D. Independent Study:  

INSTRUCTIONAL TECHNIQUES:  
Lecture and/or discussion on specific topics with sample code demo using projector and white board.  

COURSE ASSIGNMENTS:  
Reading Assignments  
Text & Websites  
Out-of-class Assignments  
An optional library research paper will promote further study and research in current Java programming or other related topics selected by the student and approved by the instructor.  

Writing Assignments  
Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their projects.  

METHODS OF STUDENT EVALUATION:  
Midterm Exam  
Final Exam  
Short Quizzes  
Written Assignments  
Essay Examinations
Objective Examinations
Projects (ind/group)
Problem Solving Exercises
Oral Presentations
Skills Demonstration

**Demonstration of Critical Thinking:**

Students will be demonstrating their laboratory projects. Optional research papers and classroom presentations will further demonstrate their ability in critical thinking and problem solving.

**Required Writing, Problem Solving, Skills Demonstration:**

Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their projects.

**TEXTS, READINGS, AND RESOURCES:**

**TextBooks:**

**Other:**
1. A syllabus, and multiple reference material related to the latest software development technologies will be distributed by the instructor.

**LIBRARY:**

Adequate library resources include: Print Materials

**Comments:**

Adequate

**Attachments:**

[Attached Files]