

**SAN DIEGO COMMUNITY COLLEGE DISTRICT
CITY, MESA, AND MIRAMAR COLLEGES
ASSOCIATE DEGREE COURSE OUTLINE**

SECTION I**SUBJECT AREA AND COURSE NUMBER:** Computer and Information Sciences 189A**COURSE TITLE:** Introduction to Programming I**Units: 4**
Grade Only**CATALOG COURSE DESCRIPTION:**

Using the popular programming language Java, this course introduces students to the process of developing simple software applications to solve typical human problems. This includes language syntax, structure, and semantics as well as the basics of object-oriented software engineering. CISC 189A and B together are a slower-paced version of CISC 190, with more programming practice. CISC 189A is the first of the two-course sequence.

REQUISITES:**Advisory:**

CISC 150 with a grade of "C" or better, or equivalent

or

CISC 181 with a grade of "C" or better, or equivalent

&

ENGL 049 with a grade of "C" or better, or equivalent or Assessment Skill Level W5

&

MATH 046 with a grade of "C" or better, or equivalent or Assessment Skill Level M40

and

Limitation on Enrollment:

This course is not open to students with previous credit for CISC 190-Java Programming

FIELD TRIP REQUIREMENTS: May be required**TRANSFER APPLICABILITY:** Associate Degree Credit & transfer to CSU and/or private colleges and universities UC Transfer Course List CISC 189A and 189B are equal to 190. No credit for 189A or 189B if taken after 190 (per catalog).**TOTAL LECTURE HOURS:** 48 - 54**TOTAL LAB HOURS:** 48 - 54**STUDENT LEARNING OBJECTIVES:**

Upon successful completion of the course the student will be able to:

1. Create, compile, run, and test simple Java programs using the Java Development Kit (JDK) and an appropriate interactive development environment (IDE), such as JCreator or Eclipse.
2. Solve simple business problems with programmed software solutions, following a standard development methodology, such as a simplified version of the Unified Process, or any of the accepted agile methods. This includes understanding project stakeholders, requirements analysis, and principles

of design.

3. Apply a subset of the Unified Modeling Language (UML) diagrams to the design of their programs.

4. Apply basic Java syntax, structure, and semantics, including: console applications with Scanner input and output functions; simple algorithms and text processing and validation; classes and objects; basic graphical interaction using simple Swing components (e.g. JOptionPane); selection and repetitive statements; methods, method calls, and attributes (fields); and arrays.

5. Develop an understanding of and apply some of the key principles of object-orientation in Java, including abstraction, inheritance, and encapsulation.

SECTION II

1. COURSE OUTLINE AND SCOPE:

A. Outline Of Topics:

The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

- I. Review the basics of Information Technology literacy
 - A. Microsoft Windows Operating System
 1. Starting, stopping, and managing the system and its tools and applications
 2. Navigating the file system
 - B. Microsoft Office Word, Excel, and PowerPoint
 - C. Computer hardware and networking
 - D. Introduction to Computer Science
 1. Review of the history of programming languages
 2. Overview of principles of computer science
 - a. Language syntax and structure
 - b. Notation
 - c. Semantics and logic processing
- II. Using the Java Development Kit classes and programs in the framework of a Java Interactive Development Environment (e.g JCreator, Eclipse, ...)
 - A. Create, edit, and compile a program
 - B. Compile, test, and debug a program
- III. Create classes with attributes (fields), methods, local variables, and program logic to produce valuable functionality
- IV. Accept input from program users, process this input, and provide result output to users
 - A. Handle and validate numeric input; calculate results
 - B. Handle and validate text input
 - C. Process and format results into proper output
 - D. Use of some Swing package components
- V. Apply software development methodology best practices to the design and development of simple programs
 - A. Proper requirements acquisition and analysis
 - B. Detailed design (brief introduction to Design Patterns)
 - C. Use of simple UML diagrams to document design decisions (at least class hierarchy and activity diagrams)
 - D. Quality-centric implementation
 - E. Iterative approach - continuous improvement
 - F. Careful unit testing and error correction
 - G. Appropriate formatting and documentation (brief introduction to JavaDoc)

B. Reading Assignments:

Reading assignments are required and may include but, are not limited to, the following:

I. Weekly chapter reading assignments based on a current textbook covering Java software design and development.

II. Optionally, the instructor may use a reader, online materials, and other ancillary resources as additional reading materials.

III. Reading assignments will cover the topics described in student learning outcomes and the course

outline.

C. Appropriate Assignments that Demonstrate Critical Thinking:

Critical thinking assignments are required and may include, but are not limited to, the following: The topic of Java software programming provides ample opportunity for students to develop their problem analysis and solving skills.

I. Assignments furthering critical thinking include:

- A. project requirements analysis and validation
- B. program feature and function design
- C. program implementation and testing; problem correction and validation
- D. class discussions on software techniques and technologies
- E. research on tools and techniques used in the software industry
- F. a semester portfolio project encourages students to develop a realistic prototype of a valuable software program.
- G. in addition, instructors are encouraged to discuss the topic of software development and use in the broader context of socio-economic issues (example: access to computers for disadvantaged users)

D. Appropriate Outside Assignments:

Outside assignments may include, but are not limited to, the following:

I. Students may be asked to research course-related topics on the Internet or at the Learning Resource Center.

II. A semester-level portfolio project is recommended to help students gain practical experience with the concepts learned.

- A. The format for the portfolio includes:
 - 1. selection of a viable, realistic project
 - 2. interviews with project stakeholders to collect requirements and risks
 - 3. appropriate design specification
 - 4. implementation of a more substantial Java program that programmatically follows the design specification based on the project requirements.

III. If possible, a field trip to a local software development company is encouraged.

E. Writing Assignments:

Writing assignments are required and may include, but are not limited to, the following:

Writing assignments cover lab and homework activities. Students will be expected to hand in assignments for grading in a professional, typewritten format.

I. They fall into three categories:

- A. Java program design
 - B. Java program implementation
 - C. Java program testing and validation.
- II. Topics covered by assignments include:
- A. Java syntax, structure, and semantics, including classes and objects
 - B. methods and attributes (fields)
 - C. statements (assignment; selection; repetition)
 - D. user input and output
 - E. Java keywords
 - F. source code structuring
 - G. comments and documentation

III. Optionally, the professor may assign weekly reading summaries (especially useful in online course delivery) and/or an end-of-semester paper to cover a course-related topic.

2. METHODS OF EVALUATION:

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

- I. Students in this course will complete lab and homework assignments (on the topics listed under Topics) which will be graded.
- II. They should also be subject to at least one midterm and one final examinations.
- III. They will receive credit for class participation and team work. In addition, the assignment of a substantial semester paper and a semester or portfolio project (with presentation) is highly recommended.

3. METHODS OF INSTRUCTION:

Methods of instruction may include, but are not limited to, the following:

- * Distance Education
- * Computer Assisted Instruction
- * Audio-Visual
- * Collaborative Learning
- * Lecture-Lab Combination
- * Other (Specify)
- * Field Trip(s)

4. REQUIRED TEXTS AND SUPPLIES:

Textbooks may include, but are not limited to:

TEXTBOOKS:

1. Gary Shelly, Thomas Cashman, Joy Starks, Michael Mick. Java Programming - Comprehensive Concepts and Techniques, current ed. Thompson Course Technology, 2006, ISBN: 1418859850
2. Georges Merx and Ronald Norman. Unified Software Engineering With Java, Prentice-Hall, 2006, ISBN: 0130473766
3. Harvey Deitel and Paul Deitel. Java How To Program, current ed. Peason/Prentice-Hall, 2005, ISBN: 0131483986
4. Laura Lemay and Rogers Cadenhead. Sams Teach Yourself Java 2 Platform in 21 Days, current ed. Sams, 2004, ISBN: 0672316471
5. Walter Savitch. Java: An Introduction to Computer Science and Programming, 4th or current ed. Prentice-Hall, 2005, ISBN: 0131492020

MANUALS:

PERIODICALS:

SOFTWARE:

SUPPLIES:

1. Students need to have a removable solid-state storage device (128MB or greater) to store their program files.

ORIGINATOR: Curricunet Version 2

CO-CONTRIBUTOR(S)

DATE: 10/21/2005