

**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 189B**COURSE TITLE:** Introduction to Programming II**Units: 4**
Grade Only**CATALOG COURSE DESCRIPTION:**

Using the popular programming language Java, this course continues the process of students learning how to solve business problems by developing useful software applications. This includes more advanced concepts like abstract data structures, graphics, and data persistence. CISC 189A and B together are a slower-paced version of CISC 190, with more programming practice. CISC 189B is the second of the two-course sequence.

REQUISITES:**Prerequisite:**

CISC 189A with a grade of "C" or better, or equivalent
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. Build on CISC 189A to apply additional principles of object orientation, such as polymorphism and the application of Design Patterns
2. Develop multi-class solutions that implement inheritance, composition, and interfaces
3. Implement graphical applications with swing, including layout managers and event handlers/action listeners
4. Make information persistent by implementing file input-output operations
5. Develop programs that implement exception handling and input validation, including the use of regular expression and pattern matching
6. Apply practical principles of object-oriented software development methodology Best Practices to the design, development, and testing of useful applications

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. Develop programs that use more than one class
 - A. Use sub-and super-classes to implement inheritance
 - B. Apply composition to create objects from other objects and pass information as parameters and return values
 - C. Apply encapsulation to properly control access to object attributes
- II. Develop class hierarchies that implements polymorphism
 - A. Overload methods and constructors
 - B. Implement type polymorphism across sub/super class structure
- III. Develop a multi-class solution that uses swing components, including JButtons, JFrames, etc. and uses the ActionListener interface to handle action events
- IV. Develop your own interface and use it in a program
- V. Implement input validation
 - A. Use exception handling to test for numeric input
 - B. Use pattern matching to recognize and accept properly formatted input
 - C. Use string and character handling methods to parse strings
- VI. Create persistent data files using Java file input-output classes
 - A. Use file I/O exception handling
 - B. Optionally/alternatively, use Java database connectivity to store data in a relational database
 1. Apply SQL statements
 2. Use database I/O exception handling
- VII. Develop a larger multi-class portfolio project that include requirements analysis, design, implementation, and testing phases
- VIII. Preview more advanced Java technologies, such as handling of multimedia; networking; and distributed computing

B. Writing Assignments:

Writing assignments are required and may include, but are not limited to, the following: Students will be expected to hand in assignments for grading in a professional, typewritten format. Assignments that include source code should also have program screen shots and sufficient explanatory information to document the work properly and completely.

- I. Writing assignments cover lab and homework activities. 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 and associated class hierarchies
 - B. Object-orientation in Java, e.g. abstraction; inheritance; encapsulation; composition; and polymorphism
 - C. Abstract data structures
 - D. Object persistence
 - E. Graphical user interfaces with layout managers and event handlers
- 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.

C. Appropriate Assignments that Demonstrate Critical Thinking:

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

- I. The topic of Java software programming provides ample opportunity for students to develop their problem analysis and solving skills.

- II. 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.
- III. 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. 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.

E. 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 at the Learning Resource Center.
- II. A semester-level portfolio project is recommended to help students gain practical experience with the concepts learned.
- III. The format for the portfolio includes:
 - A. selection of a viable, realistic project
 - B. interviews with project stakeholders to collect requirements and risks
 - C. appropriate design specification
 - D. implementation of a more substantial Java program that programmatically follows the design specification based on the project requirements.
- IV. If possible, a field trip to a local software development company is encouraged to learn about real-life software development.

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. Georges Merx and Ronald Norman. Unified Software Engineering With Java, current ed. Prentice-Hall, 2006, ISBN: 0130473766
2. Harvey Deitel and Paul Deitel. Java How To Program, current ed. Pearson/Prentice-Hall, 2005, ISBN: 0131483986
3. Mark Allen Weiss. Data Structures and Problem Solving Using Java, current ed. Pearson/Addison-Wesley, 2006, ISBN: 0321322134
4. Michael Main. Data Structures and Other Objects Using Java, current ed. Pearson/Addison-Wesley, 2006, ISBN: 0321375254
5. Walter Savitch. Java: An Introduction to Computer Science and Programming, 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: 10/25/2005