



CIS 255 (CS1) Programming Methods: Java (4) Continuation of CIS 254. Object-oriented programming methodology for both computer science majors and computer professionals. Systematic approach to design, construction, and management of computer programs; emphasizing program documentation, testing, debugging, maintenance and software reuse. Also includes UML, virtual machines, exception handling, sorting and searching algorithms, recursion, inheritance, polymorphism, fundamental graphics, and computer ethics. This course conforms to the ACM CS1 standards. A materials fee in the amount shown in the Schedule of Classes is payable upon registration. Prerequisite: MATH 120 or equivalent; CIS 254 or equivalent. Recommended preparation: Eligibility for ENGL 838/848. Pass/No Pass or letter grade option. (AA: Area C1/Area E2c, CSU, UC)

**6. Student Learning Outcomes (Identify 1-6 expected learner outcomes using active verbs.)**

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

1. Demonstrate knowledge and understanding of the principal object-oriented programming concepts.
2. Design, implement, and use classes, interfaces, and methods, employing standard naming conventions and advanced features including exception handling, I/O, GUIs, and event handling.
3. Employ object-oriented methodology to design and effectively implement medium-sized computer programs using simple Unified Modeling Language (UML) notation.
4. Decompose a real-world problem and apply strategies for the reuse of existing components with inheritance and polymorphism.
5. Describe the concept of recursion, and implement, test, and debug simple recursive methods.
6. Explain and employ basic sorting and searching algorithms.
7. Use and create standard API documents to understand and document the use of classes and methods.
8. Demonstrate an understanding of professional codes of ethics, such as ACM and IEEE.

**7. Course Objectives (Identify specific teaching objectives detailing course content and activities. For some courses, the course objectives will be the same as the student learning outcomes. In this case, "Same as Student Learning Outcomes" is appropriate here.)**

Same as Student Learning Outcomes

**8. Course Content (Brief but complete topical outline of the course that includes major subject areas [1-2 pages]. Should reflect all course objectives listed above. In addition, a sample course syllabus with timeline may be attached.)**

See attached topical outline

**9. Representative Instructional Methods (Describe instructor-initiated teaching strategies that will assist students in meeting course objectives. Describe out-of-class assignments, required reading and writing assignments, and methods for teaching critical thinking skills. If hours by arrangement are required, please indicate the additional instructional activity which will be provided during these hours, where the activity will take place, and how the activity will be supervised.)**

The course will include the following instructional methods as determined appropriate by the instructor:

- Lecture will be used to introduce new topics;
- Teacher will model problem-solving techniques;

- Class will solve a problem together, each person contributing a potential "next step";
- Students will participate in short in-class projects (in teacher-organized small groups) to ensure that students experiment with the new topics in realistic problem settings;
- Teacher will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding;
- Teacher will create and manage an Internet conference for discussion of course topics; and
- Students will work in small groups to solve programming assignments.

**10. Representative Methods of Evaluation** (Describe measurement of student progress toward course objectives. Courses with required writing component and/or problem-solving emphasis must reflect critical thinking component. If skills class, then applied skills.)

- Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher on objectives 1 - 6;
- Assessment of student contributions during class discussion and project time to assess objectives 1-5;
- Individual programming assignments to assess objectives 1, 3, 4, 5, and 7;
- Midterm and Final exams (short answer (textbook material), general problem solving (similar to in-class work), short program segments (similar to programming assignments) ) to assess objectives 1-6, 8;
- Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort to assess objectives 3-4.

**11. Representative Text Materials** (With few exceptions, texts need to be current. Include publication dates.)

Java: How to Program, Deitel, Pearson ISBN 0-13-605306-8, 2010

Introduction to Java Programming, Comprehensive Version, Liang, Pearson ISBN 0-13-601267-1, 2009

Introduction to Programming Using Java, Eck, <http://math.hws.edu/javanotes/>, 2009

Prepared by: \_\_\_\_\_  
(Signature)

Email address: greenm@smccd.edu

Submission Date: \_\_\_\_\_