INSTRUCTOR  Melissa Green  greenm@smccd.edu  http://www.smccd.edu/accounts/greenm
Office: 17-148  650-574-6374  Office hours: MWTh 3:30-4:30 PM  T 4:00-6:00 PM

LECTURE  CIS 255 HY  CRN 42555  4 Units  Thursday, 5:00-7:40 PM  Room 17-105
LAB  ONLINE
Plus One Hour by Arrangement
Students will use the textbook’s CD-ROM, textbook web site, the Sun Java API and the World Wide Web to enrich their learning.


COURSE DESCRIPTION
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, fundamental graphics, and computer ethics. This course conforms to the ACM CS1 standards.

Prerequisites:  Math120 (Intermediate Algebra) or equivalent, CIS 118/119 or CIS 254 (Introduction to Object-Oriented Program Design) or equivalent. Please note that these are serious prerequisites. CIS 255 is NOT intended as a first course in programming.

GRADING
Tests 45%  90-100 A  80-89 B
Final Exam 25%  70-79 C  60-69 D
Assignments 30%  0-59 F

Students with a cumulative grade one point below cutoff will be promoted to the next highest grade if they have completed all assignments.

This course does allow “pass/no pass” grading. You must maintain a “C” average to pass.

There will be approximately 8 programming assignments as well as lab assignments. Programming assignments will be graded on program correctness, documentation, and style. There will be 4 tests over the semester, each worth 75 points. Each test focuses on recent material but may also cover material from the beginning of the semester. The tests will be based on the textbook, handouts, and techniques you have used on related assignments. I will use your 3 highest test scores in determining your test grade. There are NO makeup tests.

OTHER IMPORTANT DATES
Monday, February 1, 2010  Last day to add or drop with eligibility for fee credit or refund
Sunday, February 7, 2010  Last day to complete WebSMART registration
Tuesday, February 16, 2010  Last day to drop classes with no notation on student record
Tuesday, February 23, 2010  Last day to declare pass/no pass option
March 10-12, 2010  Flex Days – NO CLASS
April 3-9, 2010  Spring Break – NO CLASS
Thursday, April 29, 2010  Last day to withdraw with a “W” on student record
COMPLETING ASSIGNMENTS
This course will require at least twelve hours of computer work each week in addition to preparation time. All assignments must be uploaded to WebAccess by the due date/time. In addition, you must submit a printout of all source code for each assignment. Assignments will NOT be accepted by e-mail. Students are expected to do their own work. Any case of duplicate assignments will result in a grade of zero for all people involved. Late assignments will have a 50% penalty and are accepted only up to the beginning of the next class. You will receive a separate handout with programming guidelines.

PARTICIPATION
Under normal circumstances I do NOT drop students from the class rolls. It is the student’s responsibility to file the paperwork needed to drop or withdraw from this class. If you stop participating in the class you will probably receive an “F”.

FINAL EXAM SCHEDULE
The final exam covers all material for the semester.
CIS 255 HY  Thursday, May 27, 2010  5:00-7:30 PM

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