

California University of Pennsylvania
Guidelines for New Course Proposals
University Course Syllabus

Department of Applied Engineering and Technology

A. Protocol

Course Name: Technical Computing using Java
Course Number: CET350
Credits: 3
Prerequisites: CSC 265 Object Oriented Programming,
MAT281 Calculus I
Revision Date: 6-Mar-2006

B. Objectives of the Course

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

- a) Demonstrate how Java can be used to solve real life problems.
- b) Employ various Java methods in computational solutions.
- c) Formulate and apply Java applications to solve a given problem.
- d) Formulate and apply Java Applets to solve a given problem.
- e) Design and employ user interaction routines for Java programs.
- f) Design and employ file interaction routines for Java programs.
- g) Design and employ graphical interaction routines for Java programs and applets.
- h) Identify and discuss components of object-oriented programming as implemented in the Java language.
- i) Create Java programs that interface and interact with system hardware components.

C. Catalog Description

This course enables the student to acquire a thorough understanding of the Java language and its application in solving real world problems. Both Java programs and Applets will be studied. Emphasis is place on efficient software development using structured programming techniques. Students are required to design, write, test, and run programs using an appropriate version of Java. Prerequisites: CSC 265 Object Oriented Programming, MAT 281 Calculus I. 3 credits.

D. Outline of the Course

1. Review Object Oriented Programming concepts
 - Data
 - a. types
 - b. classes
 - i. wrappers
 - ii. number methods
 - iii. conversion methods
 - iv. comparison methods
 - c. conversion
 - d. kind
 - 2. Operators
 - a. mathematical
 - b. relational
 - c. logical
 - d. increment/decrement
 - e. application of operators to numerical computing
 - 3. Statements
 - a. invocation
 - b. assignment
 - c. repetition
 - d. selection
 - e. exception

- f. application of program control flow in engineering problem solving
- 4. Methods
 - a. modifiers
 - b. kind
 - c. declaring
 - d. calling
 - e. parameter passing
- 5. Classes
 - a. fields
 - b. methods
 - c. inheritance
- 6. Objects
 - a. instantiation
 - b. access
 - c. protection
 - d. hierarchy
 - e. overriding methods
 - f. scientific computing using object-oriented design
- 7. Input/Output
 - a. console
 - b. GUI
- 8. Arrays, Buffers, Stacks, Strings, and Vectors
 - a. creating data structures in Java
 - b. using data structures in Java
 - c. application of data structures in technical computing
- 9. Files
 - a. testing
 - b. modifying
 - c. opening
 - d. reading
 - e. writing
 - f. closing
- 10. GUI
 - a. components
 - b. layout managers
 - c. menus
 - d. events
 - e. images
 - f. practical applications of GUI interfaces
- 11. Applets
 - a. applications to applets
 - b. methods
 - c. html
 - d. access restrictions
 - e. web-based interfaces using Java
- 12. Threads
 - a. communication
 - b. multiple instances
 - c. priority
 - d. interfacing
 - e. signalling
 - f. use of concurrency in large or complex systems
- 13. Real-world integration
 - a. interfacing to peripheral I/O devices
 - b. applications of Java-based data communications
 - c. application of distributed computing in engineering

E. Teaching Methodology

Instruction for this course will occur using lecture, presentations, demonstrations, assignments and cooperative group methods. Each student will work in teams to produce individual modules of class projects.

F. Text

Required:

Bishop, Judith M. & Bishop, Nigel T. (2000). Java Gently For Engineers & Scientists, Addison-Wesley.

Recommended:

Chan, P., Lee, R., & Kramer, D. (1998). The Java Class Libraries Volume 1. Second Edition. Massachusetts Addison-Wesley.

Chan, P., & Lee, R. (1998). The Java Class Libraries Volume 2. Second Edition. Massachusetts Addison-Wesley.

Chan, P., Lee, R., & Kramer, D. (1999). The Java Class Libraries Second Edition, Volume 1 Supplement for the Java 2 Platform Standard Edition, V1.2. Massachusetts Addison-Wesley.

G. Assessment Activities

The final grade will be determined as a percentage from the following evaluation methods with varying weights at the discretion of the instructor:

- a) Examinations
- b) Quizzes
- c) Assignments
- d) Programs
- e) Attendance
- f) Performance

H. Accommodations for Students with Disabilities

Students with disabilities:

1. Reserve the right to decide when to self-identify and when to request accommodations.
2. Will register with the Office for Students with Disabilities (OSD) each semester to receive accommodations.
3. Will present the OSD Accommodation Approval Notice to faculty when requesting accommodations.
4. Might be required to communicate with faculty for accommodations which specifically involve the faculty.

Office for Students with Disabilities

Requests for approval for reasonable accommodations should be directed to the Office for Students with Disabilities (OSD). Approved accommodations will be recorded on the OSD Accommodation Approval notice and provided to the student. Students are expected to adhere to OSD procedures for self-identifying, providing documentation and requesting accommodations in a timely manner. The Office for Students with Disabilities is located in the Keystone Education Building – Room 112 and the phone number is (724) 938-5781.

I. Supportive Instructional Materials, e.g. library materials, internet access, etc.

Cadenhead, Rogers. (1999). Sams teach yourself Java 2 in 24 hours. Indiana Sams.

Campione, Mary. (1998). The Java tutorial : object-oriented programming for the Internet. Massachusetts Addison-Wesley.

Chan, Patrick. (1999). The Java class libraries. Massachusetts Addison-Wesley.

Deitel, Harvey M. (2000). The complete Java 2 training course the ultimate cyber classroom. New Jersey Prentice Hall.

Deitel, Harvey M. (1999). Java : how to program. New Jersey Prentice Hall.

Enete, Noel. (1997). Java jump start : a beginner's guide to Internet programming. New Jersey Prentice Hall.

- Flanagan, David. (1999). Java in a nutshell : a desktop quick reference. California O'Reilly.
- Grand, Mark. (1997). Java fundamental classes reference. California O'Reilly.
- Grand, Mark. (1997). Java language reference. California O'Reilly.
- Keogh, James Edward. (2004). Java demystified. California McGraw-Hill/Osborne.
- Ladd, Scott Robert. (1998). Java algorithms. New York McGraw-Hill.
- Lambert, Kenneth Alfred. (2000). Java : a framework for program design and data structures. California Brooks/Cole.
- Lemay, Laura. (1999). Sams teach yourself Java 2 in 21 days. Indiana Sams.
- Lemay, Laura. (1999). SAMS teach yourself Java 2 platform in 21 days. Indiana Sams.
- McCarty, Bill. (1999). Java distributed objects. Indiana Sams.
- Naughton, Patrick. (1997). Java : the complete reference. California McGraw-Hill/Osborne.
- Naughton, Patrick. (1996). The Java handbook. California McGraw-Hill.
- Nelson, Matthew T. (1998). Java foundation classes. New York McGraw-Hill.
- Niemeyer, Patrick. (1997). Exploring Java. Massachusetts O'Reilly.
- Schildt, Herbert. (2005). Java a beginner's guide. California McGraw-Hill/Osborne.
- Zukowski, John. (1997). Java AWT reference. Massachusetts O'Reilly.

Additional Information for Course Proposals

- J. Proposed Instructors: Mr. Jeffrey S. Sumey, Dr. Anthony S. Pyzdrowski
- K. Rationale for the Course
 Industry is current looking for Java programming skills in their new hires. In order to better prepare our graduates, a strong Java programming course with emphases on team-oriented design and engineering-related applications is needed. Providing a course that not only trains our students in the Java language but also prepares them for the work force by modelling their future work environment will be beneficial in securing a job after graduation.
- L. Specialized Equipment or Supplies Needed
 The Course would make use of current computer equipped laboratories.
- M. Answer the following questions using complete sentences:
1. Does the course require additional human resources? (Please explain)
 No. One section per year would be initially offered. With the removal of several upper division Computer Science courses, faculty resources would be re allocated to the new course.
 2. Does the course require additional physical resources? (Please explain)
 No. The Course would make use of current computer equipped laboratories.
 3. Does the course change the requirements in any particular major? (Please explain)
 Yes. This course will be required in the Computer Engineering Technology program.
 4. Does the course replace an existing course? (If so, list the course and attach a Course Deletion sheet)
 No.
 5. How often will the course be taught?
 Initially once per year. Future demand may require multiple sections to be offered.
 6. Does the course duplicate an existing course in another Department or College? (If the possibility exists, indicate course discipline, number, and name)

No. Java was three times twice under the CSC 485 Special Topics in Computer Science in the Spring of 1999, the Fall of 2000, and the Spring 2004. It was also used once as the language in CSC 324 Computer Graphics in the Fall 2004.

- N. If the proposed course includes substantial material that is traditionally taught in another discipline, you must request a statement of support from the department chair that houses that discipline.
- O. Please identify if you are proposing to have this course considered as a menu course for General Education. If yes, justify and demonstrate the reasons based on the categories for General Education. The UCC must consider and approve the course proposal before consideration by the General Education Committee. No consideration for a General Education menu is given at this time.
- P. Attach Approval Form.

Additional Guidelines

The following are additional guidelines that you must follow which will expedite your course proposal. Failure to follow these guidelines will result in the return of the proposal to the department.

1. Be sure that your proposal is in the correct format (Guidelines for New Course Proposals) and that all questions have been completely answered.
2. Be sure that you have completed and attached the Application to Establish a New Course form and that the appropriate signatures have been affixed.
3. Be sure that you include an updated advisement sheet for any course that is being required by the department or is classified as a restricted elective. In addition, you must include copies of the current advisement sheet(s) with your proposal. Be certain that all advisement sheets affected by the proposed course change be included with your proposal. A short memo indicating the changes to the advisement sheet(s) and rationale must be attached.
4. When submitting materials for consideration by the Curriculum Committee, you must provide 20 copies of each item to be reviewed to the Chairperson.
5. All completed items must be in the hands of the Chairperson of the Curriculum Committee a minimum of one week prior to the next regularly scheduled meeting.
6. All courses that are to be dual listed must include the rationale.
7. Any department requesting a course name change must also submit a course number change as well. Submit this request on the Application to Establish a New Course Form.
8. New advisement sheets, major proposals, minors, or changes to advisement sheets will become effective the fall semester following committee approval. The advisement sheets must also include the committee approval date at the bottom and the effective date on the advisement page. Submit this request on the Advisement and / or Program Changes form.
9. New courses will become effective the semester following committee approval.
10. Any references listed must be in the appropriate bibliographic format for the discipline.