

**Eastern Illinois University
Revised Course Proposal
MAT 2670, Computer Science II**

Please check one: New course Revised course

PART I: CATALOG DESCRIPTION

1. **Course prefix and number, such as ART 1000:** MAT 2670
2. **Title (may not exceed 30 characters, including spaces):** Computer Science II
3. **Long title, if any (may not exceed 100 characters, including spaces):** Computer Science II
4. **Class hours per week, lab hours per week, and credit [e.g., (3-0-3)]:** 3-2-4
5. **Term(s) to be offered:** Fall Spring Summer On demand
6. **Initial term of offering:** Fall Spring Summer **Year:** 2011
7. **Course description (not to exceed four lines):** Intermediate programming techniques with emphasis on object oriented design, recursion as a problem solving strategy, event-driven programming, graphical user interface design, and software engineering principles. All programming will be done in an object-oriented programming language (such as Java, C++, or Python).
8. **Registration restrictions:**
 - a. **Identify any equivalent courses** (e.g., cross-listed course, non-honors version of an honors course). No such courses
 - b. **Prerequisite(s)**, including required test scores, courses, grades in courses, and technical skills. Indicate whether any prerequisite course(s) MAY be taken concurrently with the proposed/revised course. C or better in MAT 2170
 - c. **Who can waive the prerequisite(s)?**
XX Chair XX Instructor
 - d. **Co-requisites** (course(s) which MUST be taken concurrently with this one): No co-requisites
 - e. **Repeat status:** Course may not be repeated.
 - f. **Degree, college, major(s), level, or class to which registration in the course is restricted, if any:** None
 - g. **Degree, college, major(s), level, or class to be excluded from the course, if any:** None
9. **Special course attributes** [cultural diversity, general education (indicate component), honors, remedial, writing centered or writing intensive] No special attributes
10. **Grading methods** (check all that apply): Standard letter
11. **Instructional delivery method:** lecture/lab combined

PART TWO: ASSURANCE OF STUDENT LEARNING

1. List the student learning objectives of this course: Upon completion of the course, students should be able to: organize projects using an integrated development environment, use a version control system, design, construct, and utilize classes, understand and use object oriented programming principles, understand and use object oriented design patterns, solve problems using recursion, manipulate basic data structures from the standard library of the programming language being used in the course, implement simple abstract data types, implement simple event-driven programs, graphical user interface design, understand and use UML, understand and use software engineering principles.

a. If this is a general education course, indicate which objectives are designed to help students achieve one or more of the following goals of general education and university-wide assessment:

- **EIU graduates will write and speak effectively.**
- **EIU graduates will think critically.**
- **EIU graduates will function as responsible citizens.**

n/a

b. If this is a graduate-level course, indicate which objectives are designed to help students achieve established goals for learning at the graduate level:

- **Depth of content knowledge**
- **Effective critical thinking and problem solving**
- **Effective oral and written communication**
- **Advanced scholarship through research or creative activity**

n/a

2. Identify the assignments/activities the instructor will use to determine how well students attained the learning objectives: There will be approximately 10 assignments featuring a mix of structured laboratories and programming projects. At least one midterm exam and a final will be given.

Each structured laboratory will have a graded component highlighting the current topic of lecture. The programming projects will be of sufficient size to benefit from software engineering principles and may be team based. The programming projects will also focus on the current topic of lecture.

3. Explain how the instructor will determine students' grades for the course: The final grade will be computed by a scheme similar to the following: assignments 50%, exams 20%, and final 30% (relative weights to be determined by the instructor).

4. For technology-delivered and other nontraditional-delivered courses/sections, address the following:
- Describe how the format/technology will be used to support and assess students' achievement of the specified learning objectives: n/a
 - Describe how the integrity of student work will be assured: n/a
 - Describe provisions for and requirements of instructor-student and student-student interaction, including the kinds of technologies that will be used to support the interaction (e.g., e-mail, web-based discussions, computer conferences, etc.): n/a
5. For courses numbered 4750-4999, specify additional or more stringent requirements for students enrolling for graduate credit. These include:
- course objectives; n/a
 - projects that require application and analysis of the course content; and n/a
 - separate methods of evaluation for undergraduate and graduate students. n/a
6. If applicable, indicate whether this course is writing-active, writing-intensive, or writing-centered, and describe how the course satisfies the criteria for the type of writing course identified. (See Appendix *.)
- n/a

PART III: OUTLINE OF THE COURSE

Provide a week-by-week outline of the course's content. Specify units of time (e.g., for a 3-0-3 course, 45 fifty-minute class periods over 15 weeks) for each major topic in the outline. Provide clear and sufficient details about content and procedures so that possible questions of overlap with other courses can be addressed. For technology-delivered or other nontraditional-delivered courses/sections, explain how the course content "units" are sufficiently equivalent to the traditional on-campus semester hour units of time described above.

Topics	Time
Using an IDE and a version control system	1 Week
Using a debugger	1 Week
Data and procedural abstraction	1 Week
Software engineering and UML	2 Weeks
Program correctness and efficiency	1 Week
Classes, inheritance, and design patterns	1 Week
Lists and design patterns	1 Week
Graphics and event driven programs	2 Weeks
GUI design	2 Weeks
Recursion	2 Weeks
Review and Selected Topics	1 Week

PART IV: PURPOSE AND NEED

1. Explain the department's rationale for developing and proposing the course. MAT 3870 is a course focusing on data structures. Thus, the data structure component of 2670 need not be as extensive as in the past. Its removal from 2670 and the extra lecture hour allows other topics that are fundamental to modern computer science to be taught during 2670. The added topics include: event-driven programs, graphical user interface design, UML, and software engineering principles.

a. If this is a general education course, you also must indicate the segment of the general education program into which it will be placed, and describe how the course meets the requirements of that segment. n/a

b. If the course or some sections of the course may be technology delivered, explain why. n/a

2. Justify the level of the course and any course prerequisites, co-requisites, or registration restrictions.

There will be no change in the requisites, course level and the like.

3. If the course is similar to an existing course or courses, justify its development and offering.

There is not a similar course.

a. If the contents substantially duplicate those of an existing course, the new proposal should be discussed with the appropriate chairpersons, deans, or curriculum committees and their responses noted in the proposal. n/a

b. Cite course(s) to be deleted if the new course is approved. If no deletions are planned, note the exceptional need to be met or the curricular gap to be filled. n/a

4. Impact on Program(s):

a. For undergraduate programs, specify whether this course will be required for a major or minor or used as an approved elective. MAT 2670 will remain a requirement for the Mathematics and Computer Science major and an approved elective for Group II of the Mathematics major.

b. For graduate programs, specify whether this course will be a core requirement for all candidates in a degree or certificate program or an approved elective. n/a

If the proposed course changes a major, minor, or certificate program in or outside of the department, you must submit a separate proposal requesting that change along with the course proposal. Provide a copy of the existing program in the current catalog with the requested changes noted. n/a

PART V: IMPLEMENTATION

1. Faculty member(s) to whom the course may be assigned: There will be no change in the faculty who teach the course. It will typically be taught by: Peter Andrews, Andrew Mertz, William Slough, or Nancy Van Cleave.

If this is a graduate course and the department does not currently offer a graduate program, it must document that it employs faculty qualified to teach graduate courses. n/a

2. Additional costs to students: none

Include those for supplemental packets, hardware/software, or any other additional instructional, technical, or technological requirements. (Course fees must be approved by the President's Council.) none

3. Text and supplementary materials to be used (Include publication dates): The current texts are: "The Art & Science of Java: An Introduction to Computer Science" by Eric Roberts, 2008, and "Objects, Abstraction, Data Structures and Design: Using Java" by Koffman and Wolfgang, 2004. Note that Roberts' book is also used in MAT 2170 and Koffman's book in MAT 3870.

PART VI: COMMUNITY COLLEGE TRANSFER

If the proposed course is a 1000- or 2000-level course, state either, "A community college course may be judged equivalent to this course" OR "A community college course will not be judged equivalent to this course." A community college course will not be judged equivalent to a 3000- or 4000-level course but may be accepted as a substitute; however, upper-division credit will not be awarded. A community college course may be judged equivalent to this course.

PART VII: APPROVALS

Date approved by the department or school: January 11, 2010

Date approved by the college curriculum committee: January 29, 2010

Date approved by CAA: February 18, 2010