

COURSE DESCRIPTION

Department and Course Number CMPS 453 **Course Coordinator** Andrew Lee

Course Title Intro to Software Methodology **Total Credits** 3

URL

Current Bulletin Description:

Project planning, requirement engineering, specification development techniques, structured design methods, software validation, and quality assurance. CASE tools and team dynamics. Prereq: CMPS 341, 351 both with a grade of C; or CMPS 405 and 406 both with a grade of B.

Textbook:

Alan Shalloway and James R. Trott, Design Patterns Explained: A New Perspective on Object-Oriented Design, Second Edition, Addison-Wesley, 2005.

References :

1. Kent Beck and Cynthia Andrews, Extreme Programming Explained: Embrace Change, Second Edition, Addison Wesley, 2005.
2. Rick Mugridge and Ward Cunningham, Fit for Developing Software: Framework for Integrated Tests, Prentice Hall, 2005.
3. J. B. Rainsberger, JUnit Recipes: Practical Methods for Programmer Testing, Manning, 2005.

Course Goals:

1. To introduce the students to software engineering lifecycle models.
2. To provide in-depth experience in using design patterns.
3. To introduce students to version control and configuration management.
4. To provide in-depth experience in using version control tools.
5. To introduce the students to unit testing.
6. To introduce the students to integration testing.
7. To provide hands-on experience with tools for version control, unit testing, and integration testing.
8. To provide exposure to using Extreme Programming for managing small software projects.

Course Outcomes:

1. Students will gain appreciation for software engineering methods.
2. Students will learn the significance of having 'changable' design.
3. Students will gain strong experience in using object oriented constructs and concepts.
4. Students will learn several design patterns and the benefits of using the patterns.
5. Students will learn the significance of testing driven design.
6. Students will experience using unit testing tools.
7. Students will experience using integration testing tools.
8. Students will gain in-depth experience in using version control tools.
9. Students will gain experience developing software in a team.
10. Students will gain experience use of Extreme Programming for software development.
11. Students gain experience in making technical presentation.

Prerequisites by Topic:

1. In-depth knowledge of C++ or Java and experience writing programs.
2. Exposure to object-oriented programming.
3. Data structures and algorithms.

Major Topics Covered in the Course:

1. Software Engineering Methodologies (1 lecture)
What and why of software engineering; life-cycle models.
2. Extreme Programming (2 lectures)
Agile methods; user stories; test-driven development; refactoring.
3. Configuration management and version control (2 lectures)
Workspace for software development in a team; need to control configuration and changes; version control; use of CVS.
4. Object-oriented programming (3 lectures)
Interfaces; inheritance; abstract classes; scopes of private, public, protected entities; templates; framework.
5. Design Patterns (10 lectures)
Significance of changeable design; design patterns – what and why
Details of several patterns: Facade, Adapter, Strategy, Bridge, Abstract Factory, Decorator, Observer, Template Method, Singleton, Object Pool, Factory Method.
6. Unit testing (2 lectures)
Testing individual classes; designing unit tests cases; Java Eclipse; Junit.
7. Acceptance Testing (2 lectures)
Testing from user's perspective; developing acceptance tests;
Using FIT Framework.
8. Mid-term test (1 lecture)
9. Final Exam (2 hours)

Laboratory projects (specify number of weeks on each) :

- “Typical programming assignments include:”
- Critique of a non-modular program written by the instructor. (1 week)
 - A program that can benefit from using inheritance, early in the semester, to gauge the OO skills of the students (1 week)
 - Write several programs demonstrating individual design patterns. (5 weeks)
 - Project – developing a program using Extreme Programming in team of 3-5 students. (3 weeks)
 - Presentation in the class (1 week)

Oral and Written Communications

Every student is required to submit at least _____ written reports (not including exams, tests, quizzes, or commented programs) of typically _____ pages and to make _____ oral presentations of typically _____ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

Theoretical Content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

Problem Analysis

Please describe the analysis experiences common to all course sections.

Solution Design

Please describe the design experiences common to all course sections.

See laboratory projects above.