



Course Software Engineering Fundamentals (2014-2015)

Code / Version PROG1350 (100)

Total Hours 45

Credits 3

PreRequisite(s)

CoRequisite(s)

Course Description

Standard software development methodologies, testing methodologies, and software maintenance topics will be covered to prepare students for real life software development. The course will emphasize proper software design, software development issues, and documentation. The importance of effective software engineering documentation and effective technical communications cannot be stressed enough, especially in today's Rapid Application Development world. Topics include specification writing for system development and testing, code development, testing methodologies (user interface testing, regression testing, etc.), and software maintenance issues.

PLAR Eligible: Yes

Course Outcomes

Successful completion of this course will enable the student to:

1. Explain the steps in effective software and system design.
2. Use team-based approaches to designing, developing and discussing systems.
3. Write a Software Requirements Specification according to established standards.
4. Explain the importance of formal design before coding.
5. Explain the challenges and approaches used in software design.
6. Explain the importance of properly specifying testing procedures to ensure software quality.
7. Test software and learn to effectively document bugs and issues.
8. Explain the basics behind intelligent user interface design, including paper mockups and prototypes.
9. Explain debugging techniques, including assertions, conditional compilation and effective logging.
10. Explain software versioning systems and the use of revision control (RCS) in software maintenance. Use an RCS to create and control different versions of software.
11. Explain the possible consequences of faulty design processes.

Essential Employability Skills addressed in this course			Taught	Reinforced	Assessed
Communication	ⁿ	Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience	X		X
	ⁿ	Respond to written, spoken, or visual messages in a manner that ensures effective communication		X	
Numeracy	ⁿ	Execute mathematical operations accurately			
Critical Thinking and Problem Solving	ⁿ	Apply a systematic approach to solve problems	X		X
	ⁿ	Use a variety of thinking skills to anticipate and solve problems		X	X
Information Management	ⁿ	Locate, select, organize, and document information using appropriate technology and information systems		X	X
	ⁿ	Analyze, evaluate, and apply relevant information from a variety of sources		X	X
Interpersonal	ⁿ	Show respect for the diverse opinions, values, belief systems, and contributions of others		X	



Course Software Engineering Fundamentals (2014-2015)

Code / Version PROG1350 (100)

Essential Employability Skills addressed in this course			Taught	Reinforced	Assessed
Interpersonal	ⁿ	Interact with others in groups or teams in ways that contribute to effective working relationships and the achievement of goals		X	
Personal	ⁿ	Manage the use of time and other resources to complete projects		X	
	ⁿ	Take responsibility for one's own actions, decisions, and consequences		X	

Unit Outcomes

Successful completion of the following units will enable the student to:

1.0 Software Design Basics

- 1.1 Explain the importance of software design.
- 1.2 Prepare software design documentation.
- 1.3 Explain the cost of designing incorrectly.

2.0 Software Requirements Analysis

- 2.1 Explain the importance of effective communication with the customer.
- 2.2 Create a Software Requirements Specification.

3.0 User Interface Design Techniques

- 3.1 Explain and analyze various interfaces according to basic User Interface Design Philosophy and Guidelines.
- 3.2 Prototype user interfaces.

4.0 Software Design

- 4.1 Explain the challenges faced in Software Design.
- 4.2 Explain and demonstrate the desirable characteristics of software design.
- 4.3 Compare and contrast the different approaches used in software design.

5.0 Testing and Debugging Software

- 5.1 Explain the Scientific Approach to Debugging.
- 5.2 Explain the use of conditional compilation, assertions and effective logging techniques.
- 5.3 Explain and demonstrate effective bug-tracking skills.

6.0 Software Quality Assurance

- 6.1 Describe the concept of Software Quality.
- 6.2 Describe the various levels and types of software testing.
- 6.3 Explain the benefits and need of having proper Testing Specifications.
- 6.4 Explain the need for quality in mission-critical and regulated systems.

7.0 Software Maintenance

- 7.1 Explain various version numbering systems.
- 7.2 Explain the need for custom revisions of software.
- 7.3 Explain configuration and revision management.
- 7.4 Use a revision control system.
- 7.5 Use documentation generation tools.



Course Software Engineering Fundamentals (2014-2015)

Code / Version PROG1350 (100)

8.0 Object-Oriented Programming

8.1 Describe the basic principles behind object-oriented programming.

9.0 Risks of Faulty Design

9.1 Examine and evaluate real-life examples of design and process problems.

10.0 Teamwork

10.1 Work within a team.

Required Student Resources

Steve McConnell. Code Complete 2 (2004). Microsoft Press.

Class notes will be provided as required.

Optional Student Resources

Evaluation

The minimum passing grade for this course is 55 (D).

In order to successfully complete this course, the student is required to meet the following evaluation criteria:

Mid-Term Exam	20.00
Final Exam	20.00
Assignment #1	20.00
Assignment #2	15.00
Assignment #3	10.00
In-Class Task #1	5.00
In-Class Task #2	10.00
	<hr/>
	100.00 %

Other

Conestoga College is committed to providing academic accommodations for students with documented disabilities. Please contact the Accessibility Services Office.

Prepared By Sean F. Clarke

School Information Technology

Date 2014-08-13

© Conestoga ITAL