

ITT Technical Institute
CS240
Software Development Lifecycles
Onsite Course

SYLLABUS

Credit hours: 4

Contact/Instructional hours: 50 (30 Theory Hours, 20 Lab Hours)

Prerequisite(s) and/or Corequisite(s):

Prerequisite: CS200 Programming in Java I or equivalent

Course Description:

This course covers the concepts and soft skills needed to be functional on a software development team. Some requirements gathering and design techniques are also covered. The purpose of the course is to provide students with insight into the software development process in the workplace.

Syllabus: Software Development Lifecycles

Instructor:	_____
Office hours:	_____
Class hours:	_____

Major Instructional Areas

1. Fundamentals of software engineering
2. Team dynamics
3. Object modeling using UML
4. Requirements specification and analysis
5. Product design
6. User interface design
7. Coding practices and documentation
8. Testing
9. Project management
10. Design patterns

Course Objectives

1. Explain the goals of software engineering.
2. Demonstrate the ability to participate as a member of a project team.
3. Describe popular software process models.
4. Use Uniform Modeling Language (UML) to create an object model.
5. Analyze business requirements to create a requirements specification.
6. Design the interactions between processes in a system.
7. Design the user interface of a system.
8. Explain the characteristics of various implementation approaches.
9. Exercise good coding and documentation practices.
10. Apply the principles of object-oriented testing.
11. Explain the significance and characteristics of various project management documents.
12. Apply design patterns to common programming problems.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Participate as a member of a team.

2. Exercise leadership.
3. Allocate time.
4. Allocate human resources.
5. Allocate money.
6. Select and analyze information and communicate the results.
7. Determine which set of procedures will produce the desired results and make clear recommendations including rationale.
8. Analyze system and develop new or alternative systems.
9. Demonstrate the ability to make a rational decision based on analysis of accepted theories, evidence and logical thinking.
10. Demonstrate the ability to effectively and efficiently use the ITT Tech Virtual Library.

Course Outline

Note: All graded activities, except the Project, are listed below in the pattern of <Unit Number>.<Assignment Number>. For example, Labs: 2.1 refers to the 1st lab activity in Unit 2.

Unit	Activities
1—Introduction to Software Engineering	<ul style="list-style-type: none"> • Content Covered: <ul style="list-style-type: none"> <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 1, “Introduction to Software Engineering” • Labs: 1.1 • Assignments: 1.1
2—Object-Oriented Paradigm	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 2, “Object-Oriented Paradigm Overview” • Read from <i>Solid Code: Optimizing the Software Development Life Cycle</i> <ul style="list-style-type: none"> Chapter 1, “Code Quality in an Agile World” • Labs: 2.1 • Assignments: 2.1 • Project: Part 1.1 due
3—Object-Oriented Analysis	<ul style="list-style-type: none"> • Read from: <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 3, “Object-Oriented Analysis” • Labs: 3.1 • Assignments: 3.1 • Project: Part 2.1 due
4—Product Design	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 4, “Product Design” • Labs: 4.1 • Assignments: 4.1 • Project: Part 3.1 due • Quizzes: 4.1
5—Class Design	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 5, “Class Design” • Labs: 5.1 • Assignments: Part 4.1 due • Project: 5.1
6—Design Case Study	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering:</i> <ul style="list-style-type: none"> ○ Chapter 6, “CASE STUDY: Game2D with Method Design” • Labs: 6.1 • Assignments: 6.1 • Project: Part 5.1 due • Exam I

Unit	Activities
7—Implementation and Testing	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering</i>: <ul style="list-style-type: none"> ○ Chapter 7, “Implementation” ○ Chapter 8, “Testing” • Labs: 7.1 • Assignments: 7.1 • Project: Part 6.1 due
8—Project Management	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering</i>: Chapter 9, “Project Management” • Labs: 8.1 • Assignments: 8.1 • Project: Part 7.1 due • Quizzes: 8.1
9—Design Patterns and Preparing the Presentation	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering</i>: <ul style="list-style-type: none"> ○ Chapter 10, “Design Patterns” ○ Chapter 12, “Completing and Presenting the CLASS PROJECT” • Labs: 9.1 • Assignments: 9.1 • Project: Part 8.1 due
10—Horror Stories and Presentations	<ul style="list-style-type: none"> • Read from <i>Project-Based Software Engineering</i>: <ul style="list-style-type: none"> ○ Chapter 11, “Software Development Horror Stories” • Labs: 10.1 • Assignments: 10.1 • Project: Part 9.1 Presentations due
11—Review and Exam	<ul style="list-style-type: none"> • Review • Exam II

Instructional Methods

This course introduces many of the procedures you will encounter when working on a software development team. You will learn the fundamental concepts for each phase of the software development lifecycle (SDLC) as well as basic object-oriented design principles, including requirements analysis, implementation strategies, and testing. You will also learn about the role of a project manager.

The key to a successful course is to encourage team dynamics. In the project, you and other students will work together through each phase of requirements gathering, design, and prototyping. During Unit 10, you will present your project to the class.

Some activities are structured to allow you to create a design or program flow and critique the designs and program flows of other students. These critiques are important to help you learn how to give and receive constructive criticism.

Hands-on practice is an essential part of this course. Labs allow you to create various types of UML diagrams, project management documents, and other types of documents. Some labs also ask you to write code using Java to practice good coding practices, such as documenting code.

Writing assignments give you a chance to explain concepts and work on your writing skills. You will also have the opportunity to prepare and give PowerPoint presentations about various software development and project management issues.

Instructional Materials and References

Stiller, Evelyn. *Project-Based Software Engineering: An Object-Oriented Approach*. Boston: Addison Wesley, 2002.

Other Required Resources

In addition to the student textbook package, the following is also required in this course:

- Marshall, Donis, and John Bruno. *Solid Code: Optimizing the Software Development Life Cycle*. Redmond, WA: Microsoft Press, 2009.

Available in the ITT Tech Virtual Library: Main Menu> Books> Books 24x7

Equipment and Tools

- Windows XP Professional Service Pack 2
- Microsoft Visio
- Microsoft Project
- Microsoft Office
- Java Platform, Standard Edition 6.0 or Java Platform, Enterprise Edition 6.0

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at <http://library.itt-tech.edu/> to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click “Books” or use the Library Catalog on the home page to find the following books.

ITT Tech Virtual Library> Main Menu> Books> Books 24x7

- Andersson, Eve, Philip Greenspun, and Andrew Grumet. *Software Engineering for Internet Applications*. Cambridge, MA: The MIT Press, 2006.
- Biafore, Bonnie. *Visio 2007 Bible*. Indianapolis: John Wiley & Sons, 2007.
- Brandon, Dan. *Project Management for Modern Information Systems*. Hershey, PA: IGI Publishing, 2006.
- Chatfield, Carl S., and Timothy D. Johnson. *Microsoft Office Project 2007 Step by Step*. Redmond, WA: Microsoft Press, 2007.
- Dennis, Alan R., Barbara Haley Wixom, and David Tegarden. *Systems Analysis and Design with UML Version 2.0 - An Object Oriented Approach, Third Edition*. Hoboken, NJ: John Wiley & Sons, 2005.
- Dennis, Alan R., Barbara Haley Wixom, and Roberta M. Roth. *Systems Analysis and Design, Fourth Edition*. Hoboken, NJ: John Wiley & Sons, 2009.
- Ghaoui, Claude. *Encyclopedia of Human Computer Interaction*. Hershey, PA: IGI Publishing, 2006.
- Harris, Paul Eastwood. *Planning and Scheduling Using Microsoft Office 2007, Revised 2009: Including Microsoft Project 2000 to 2003*. Victoria, Australia: Eastwood Harris Pty Ltd., 2007.
- Hunt, John. *Guide to the Unified Process Featuring UML, Java, and Design Patterns*. New York: Springer, 2003.
- Jalote, Pankaj. *An Integrated Approach to Software Engineering*. 3rd ed. New York: Springer, 2005.
- Jonasson, Hans. *Determining Project Requirements*. Boca Raton, FL: Auerbach Publications, 2008.
- Keyes, Jessica. *Leading IT Projects: The IT Manager's Guide*. Boca Raton, FL: Auerbach Publications, 2009.
- Kuchana, Partha. *Software Architecture Design Patterns in Java*. Boca Raton, FL: Auerbach Publications, 2004.
- Lecky-Thompson, Guy W. *Corporate Software Project Management*. Hingham, MA: Cengage Charles River Media, 2005.
- Leon, Alexis. *A Guide to Software Configuration Management*. Boston: Artech House, 2000.
- Lewis, William E. *Software Testing and Continuous Quality Improvement, Third Edition*. Boca Raton, FL: Auerbach Publications, 2009.
- Marmel, Elaine. *In the Trenches with Microsoft Office Project 2007*. Redmond, WA: Microsoft Press, 2009.

- Marshall, Donis, and John Bruno. *Solid Code: Optimizing the Software Development Life Cycle*. Redmond, WA: Microsoft Press, 2009.
- McManus, John. *Risk Management in Software Development Projects*. St. Louis: Butterworth-Heinemann, 2004.
- Munson, John C. *Software Engineering Measurement*. Boca Raton, FL: Auerbach Publications, 2003.
- Page, Alan, Ken Johnston, and B.J. Rollison. *How We Test Software at Microsoft*. Redmond, WA: Microsoft Press, 2009.
- Peckham, Joan, and Scott J. Lloyd. *Practicing Software Engineering in the 21st Century*. Hershey, PA: IGI Publishing, 2003.
- Peters, Lawrence J. *Getting Results from Software Development Teams*. Redmond, WA: Microsoft Press, 2008.
- Stover, Teresa S., and Bonnie Biafore. *Microsoft Office Project 2007 Inside Out*. Redmond, WA: Microsoft Press, 2007.
- Syed, Mahbubur Rahman, and Sharifun Nessa Syed. *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*. Hershey, PA: IGI Global, 2009.
- Taylor, James. *Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives*. New York: AMACOM, 2004.
- Thayer, Richard H., and Mark J. Christensen. *Software Engineering, Volume 1: The Development Process*. 3rd ed. Hoboken, NJ: John Wiley & Sons, 2005.
- Tomayko, James E., and Orit Hazzan. *Human Aspects of Software Engineering*. Hingham, MA: Charles River Media, 2004.
- Van Vliet, Hans. *Software Engineering: Principles and Practice*. 3rd ed. Indianapolis: John Wiley & Sons, 2008.

ITT Tech Virtual Library> Main Menu> Books> NetLibrary

- Lewis, James P. *Fundamentals of Project Management. WorkSmart Series*. 3rd ed. New York: AMACOM Books, 2007.
- Westland, Jason. *The Project Management Life Cycle: A Complete Step-by-Step Methodology for Initiating, Planning, Executing & Closing a Project Successfully*. Philadelphia: Kogan Page, 2006.

Reference Resources

You may click "Reference" or use the E-Journal Lookup on the home page to find the following reference resource.

Reference> Additional reference resources: Project Management

- Microsoft Project
A tour, list of features, and FAQs about Microsoft Project

Other References

The following resources may be found **outside** of the ITT Tech Virtual Library.

Web sites

- ITtoolbox Project Management Knowledge Base
<http://projectmanagement.ittoolbox.com/> (accessed Jan. 4, 2010)
An online community in which peers share knowledge about information technology
- Microsoft Developer Network
<http://msdn.microsoft.com> (accessed Feb. 22, 2010)

- A library of programmer documentation for Visual Studio, SQL Server, and other Microsoft platforms
- Microsoft Office Project
<http://www.microsoft.com/project/en/us/default.aspx>
 (accessed Feb. 22, 2010)
 Help page from Microsoft; contains product documentation and training presentations for Microsoft Project
 - Microsoft Office Visio
<http://office.microsoft.com/en-us/visio/FX100649211033.aspx?CTT=96&Origin=CL100636311033> (accessed Feb. 22, 2010)
 Help page from Microsoft; contains product documentation and training presentations for Visio
 - Sun Java Developer Network
<http://java.sun.com> (accessed Feb. 22, 2010)
 A library of programmer documentation for Java
 - The Value of Project Lifecycle Methodology
<http://www.lifecyclestep.com/open/401.0HomeValue.htm> (accessed Jan. 4, 2010)
 A site specializing in developing, consulting, and training in business methodologies
 - Wideman Comparative Glossary of Project Management Terms
<http://www.maxwideman.com/pmglossary/> (accessed Jan. 4, 2010)
 Provides definitions for commonly used terms in project management

All links to Web references outside of the ITT Tech Virtual Library are always subject to change without prior notice.

Course Evaluation and Grading

Evaluation Criteria Table

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Quizzes	10%
Exam I	20%
Exam II	20%
Assignments	15%
Labs	15%
Project	20%
Total	100%

Note: Students are responsible for abiding by the Plagiarism Policy.

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

A	90–100%	4.0
B+	85–89%	3.5
B	80–84%	3.0
C+	75–79%	2.5
C	70–74%	2.0
D+	65–69%	1.5

D	60–64%	1.0
F	<60%	0.0

(End of Syllabus)