

# **CS290T**

## **Software Development Capstone Project**

### **[Onsite]**

#### **Course Description:**

This course provides an opportunity for the student to synthesize the theories and practices covered in the entire program by analyzing, designing and completing a software application development project. Teamwork, project management and presentation skills will also be integrated into the project.

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

Prerequisites: Completion of a minimum of 80 credits earned in the program of study including IT219T Program in Java II or equivalent and CS220T Server-Side Web Programming or equivalent

**Credit hours: 4**

**Contact hours: 60 (36 Theory Hours, 24 Lab Hours)**

# Syllabus: Software Development Capstone Project

---

Instructor: \_\_\_\_\_

Office hours: \_\_\_\_\_

Class hours: \_\_\_\_\_

## Major Instructional Areas

1. Project management
2. Software development lifecycle
3. Web application programming
4. Windows application programming
5. Database design, implementation, and deployment
6. System testing and deployment
7. Team dynamics
8. Documentation
9. Presentation skills

## Course Objectives

1. Apply project management concepts and techniques to a software development project.
2. Use Microsoft Project to help plan and manage a software development project.
3. Determine the requirements for a medium-sized development project.
4. Apply the knowledge acquired in the program to design an effective software solution for a given problem.
5. Apply knowledge acquired in the program to implement an effective software solution for a given problem.
6. Apply the knowledge acquired in the program to test the solution.

7. Work in teams on a medium-scope project.
8. Create the necessary documentation for the software solution.
9. Deliver a professional presentation on the design and functionality of the software solution.
10. Complete a comprehensive skills assessment for the program of study.

## 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. Maintain information to maximize retention and expression of knowledge.
2. Work effectively as part of a team and contribute ideas, suggestions, and effort to the group.
3. Identify how technological systems work and operate effectively.
4. Demonstrate competence in understanding systems.
5. Use interpersonal skills and communication techniques.
6. Demonstrate competence in selecting a technology, including determining the desired outcomes and applicable constraints.
7. Apply decision-making skills to specify goals and constraints.
8. Identify the need for data, obtain it from existing sources or create it, and evaluate its relevance and accuracy.
9. Demonstrate competence in monitoring and correcting performance.
10. Demonstrate competence in configuring, installing, and integrating various hardware and software systems.

## Course Outline

Unit	Activities
1– Introduction to the Capstone Project	<ul style="list-style-type: none"> <li>• Review project management terminology and apply the terminology to the capstone project.</li> <li>• Discuss the capstone project case studies to identify:               <ul style="list-style-type: none"> <li>○ Objectives and scope</li> <li>○ Informal scenarios</li> </ul> </li> <li>• Set up teams and define the roles and responsibilities of</li> </ul>

Unit	Activities
	<p>each team member.</p> <ul style="list-style-type: none"> <li>• Select the project from the case studies provided or create a project proposal for a different one.</li> <li>• Submit next unit: Project Documentation Part I: Project scope, informal scenarios</li> <li>• Submit next unit: Project Plan: Introduction, project organization</li> </ul>
<p>2– Requirements Specification</p>	<ul style="list-style-type: none"> <li>• Analyze requirements.</li> <li>• Create a requirements specification.</li> <li>• Create use case diagrams.</li> <li>• Prepare for structured walk-through with the instructor.</li> <li>• Submit next unit: Project Documentation Part II: Requirements specification, use case diagrams</li> <li>• Submit next unit: Project Plan: Project status report, Design schedule and task assignments</li> </ul>
<p>3– Product Design</p>	<ul style="list-style-type: none"> <li>• Perform a structured walk-through of requirements specification.</li> <li>• Select the development platform.</li> <li>• Design a persistence strategy.</li> <li>• Design the user interface.</li> <li>• Identify and plan for risks.</li> <li>• Submit next unit: Project Documentation Part III: Development platform selection, persistence strategy documentation, user interface sketches</li> <li>• Submit next unit: Project Plan: Project status report, Risk assessment</li> <li>• Submit next unit: Unit 3 status report</li> </ul>
<p>4–</p>	<ul style="list-style-type: none"> <li>• Perform system decomposition to identify classes.</li> </ul>

Unit	Activities
Class Design	<ul style="list-style-type: none"> <li>• Create interaction diagrams.</li> <li>• Create object diagrams.</li> <li>• Create a task network.</li> <li>• Create a project schedule.</li> <li>• Submit next unit: Project Documentation Part IV: Class diagram, collaboration diagrams, sequence diagrams, object diagram</li> <li>• Submit next unit: Project Plan: Project status report, Task network, project implementation schedule</li> </ul>
5– Database and Method Design	<ul style="list-style-type: none"> <li>• Check the project milestones.</li> <li>• Check the individual work and teamwork status.</li> <li>• Assess the risks to project completion and make adjustments as necessary.</li> <li>• Design the database schema.</li> <li>• Define method algorithms.</li> <li>• Create the class skeleton.</li> <li>• Create the style sheet.</li> <li>• Individual skills assessment. (This element may start earlier or later, depending on class needs. The activity will last until every individual student's interview is completed.)</li> <li>• Submit next unit: Project Documentation Part V: Class skeleton, database schema</li> <li>• Submit next unit: Project Plan: Finalized project plan, project status report</li> </ul>
6– Implementation–Part I	<ul style="list-style-type: none"> <li>• Check the project milestones.</li> <li>• Check the individual work and teamwork status.</li> <li>• Assess the risks to project completion and make adjustments as necessary.</li> </ul>

Unit	Activities
	<ul style="list-style-type: none"> <li>• Design unit tests for each method.</li> <li>• Create the database.</li> <li>• Populate the database with test data.</li> <li>• Begin implementation of the system.</li> <li>• Submit next unit: Project Documentation Part V: Unit tests</li> <li>• Submit next unit: Project Plan: Project status report</li> <li>• Submit next unit: Implementation: Database populated with some test data</li> <li>• Submit next unit: Unit 6 status report</li> </ul>
<p>7– Implementation–Part II</p>	<ul style="list-style-type: none"> <li>• Check the project milestones.</li> <li>• Check the individual work and teamwork status.</li> <li>• Assess the risks to project completion and make adjustments as necessary.</li> <li>• Complete implementation of each component of the system.</li> <li>• Perform unit tests.</li> <li>• Identify and categorize faults discovered during unit testing.</li> <li>• Submit next unit : Project Documentation Part VII: Unit test results</li> <li>• Submit next unit: Project Plan: Project status report</li> </ul>
<p>8– Implementation–Part III</p>	<ul style="list-style-type: none"> <li>• Check the project milestones.</li> <li>• Check the individual work and teamwork status.</li> <li>• Assess the risks to project completion and make adjustments as necessary.</li> <li>• Correct unit test faults and perform regression testing.</li> <li>• Design integration tests.</li> <li>• Create a bug-tracking mechanism for integration tests.</li> </ul>

Unit	Activities
	<ul style="list-style-type: none"> <li>• Identify and categorize any remaining faults discovered during unit testing.</li> <li>• Submit next unit: Project Documentation Part VII: Integration test plan</li> <li>• Submit next unit: Project Plan: Project status report</li> </ul>
<p>9–</p> <p>System Integration and Documentation</p>	<ul style="list-style-type: none"> <li>• Check the project milestones.</li> <li>• Check the individual work and teamwork status.</li> <li>• Assess the risks to project completion and make adjustments as necessary.</li> <li>• Integrate the components of the system.</li> <li>• Perform integration testing of the system.</li> <li>• Identify, categorize, and assign responsibility for correcting faults.</li> <li>• Create the user documentation.</li> <li>• Create the presentation.</li> <li>• Submit next unit : Project Documentation Part VIII: Bug reports, user documentation</li> <li>• Submit next unit: Project Plan: Project status report</li> </ul>
<p>10–</p> <p>Project Closure</p>	<ul style="list-style-type: none"> <li>• Perform acceptance testing of the system.</li> <li>• Project closure.</li> <li>• Perform a practice presentation of the solution.</li> <li>• Submit next unit: Final project documentation</li> <li>• Submit next unit: Source code</li> </ul>
<p>11–</p> <p>Final Professional Presentation</p>	<ul style="list-style-type: none"> <li>• Perform a peer review and evaluation of each team.</li> <li>• Make the final presentation.</li> </ul>



## Instructional Methods

The activities in this course simulate a real-world software development project. You will be assigned to a project team during the first unit and be presented with a case study for a real project. Throughout the course, you will perform various steps in the software development lifecycle, including designing, implementing, testing, documenting, and deploying the system. You will also perform project management tasks. You will prepare and deliver a presentation that would be used when presenting a solution to a customer for acceptance. The project will be evaluated on documentation, teamwork, presentation, and functionality of the software system.

To ensure that you acquire curriculum-related knowledge, an **individual skills assessment** will be distributed at the beginning of the course. During the course, the instructor will schedule an interview with each student in which you will be asked to answer these questions orally. The purpose of this interview is to allow you to review knowledge and skills learned in the entire program and enable you to prepare for your job interviews and career success.

The instructor—or instructors—for this course will perform the roles of advisor, consultant, and supervisor. Students in each group will perform the role of a professional consulting firm serving customers in a real-life situation. Meetings will be conducted regularly so that the instructor can provide student teams with consultation and guidance on the project and discuss various implementation and problem-solving strategies.

Complete **project documentation** in accordance with the project management guidelines is required. Grades will be based on the quality and coverage of all the areas and items listed in the Documentation Requirements section.

For **team presentation**, grades will be based on the quality in the following areas:

- Conformation to formalities—stated rules
  - Provide PowerPoint presentation demonstrating final project solution
  - Provide hand-outs for audience
  - Professional business attire mandatory
- Clarity of explanation on solution functionality and design methodology

For the **Teamwork** portion, a standard Teamwork Evaluation Form will be issued to you at the beginning of the course. Your completed form will be collected at the end of the course. Areas for evaluation will be:

- Participation
- Team organizational contributions
- Interpersonal communication performance
- Subject area expertise contributions

For the **system design**, grades will be based on how well your design documents reflect the requirements of the system and facilitate the system development. Areas for evaluation will be:

- Completeness: All required design documents should be turned in.
- Appropriateness: The design documents should identify the requirements of the system and describe how they will be addressed.

For the **system implementation** , grades will be based on the quality in the following areas:

- Functionality: Whether the system meets the functional requirements
- Usability: Whether the system includes an intuitive user interface and clear user documentation
- Maintainability: Whether the design and code is sufficiently documented to allow for modification due to changes in the business or environment

## Instructional Materials and References

### Student Textbook Package

All textbooks previously issued for the program of study

All textbooks for the core Software Development Technology courses will be used directly or indirectly as reference for this course.

### References

## ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at <http://www.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.

Books24x7>

- Andersson, Eve, Philip Greenspun, and Andrew Grumet. *Software Engineering for Internet Applications*. Cambridge, MA: The MIT Press, 2006.
- Biafore, Bonnie. *Visio 2007 Bible*. Indianapolis, IN: John Wiley & Sons, 2007.
- Chatfield, Carl S., and Timothy D. Johnson. *Microsoft Office Project 2007 Step by Step*. Redmond, WA: Microsoft Press, 2007.
- Harris, Paul Eastwood. *Planning and Scheduling Using Microsoft Office Project 2007, Revised 2009: Including Microsoft Project 2000 to 2003*. Victoria, Australia: Eastwood Harris Pty Ltd., 2009.
- Jalote, Pankaj. *An Integrated Approach to Software Engineering*. 3<sup>rd</sup> ed. New York: Springer, 2005.
- Jones, Caspers. *Estimating Software Costs: Bringing Realism to Estimating*. 2<sup>nd</sup> ed. New York: McGraw-Hill, 2007.
- Marmel, Elaine. *In the Trenches with Microsoft Office Project 2007*. Redmond, WA: Microsoft Press, 2009.
- McManus, John. *Risk Management in Software Development Projects*. Burlington, MA: Elsevier 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.
- Stover, Teresa S. *Microsoft Project 2003 Inside Out*. Redmond, WA: Microsoft Press, 2004.
- Thayer, Richard H., and Mark J. Christensen. *Software Engineering, Volume 1: The Development Process*. 3<sup>rd</sup> 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*. 3<sup>rd</sup> ed. West Sussex, England: John Wiley & Sons, 2008.

#### Books> NetLibrary

- Lewis, James P. *Fundamentals of Project Management*. WorkSmart Series. 3<sup>rd</sup> 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 resources.

- allIPM  
Articles, forums, white papers, jobs, and other project management-related content
- ganttthead.com  
Articles, news, forums, and information about the field of project management
- Microsoft Project  
A tour, list of features, and FAQs about Microsoft Project
- Project Management Institute  
The Project Management Institute (PMI) is a not-for-profit project management professional association.

## Other References

The following resources can be found **outside** of the ITT Tech Virtual Library, whether online or in hard copy.

### Web sites

- CIO.com - Business Technology Leadership  
<http://www.cio.com> (accessed January 4, 2010)  
An online IT magazine
- ITtoolbox Project Management Knowledge Base  
<http://projectmanagement.ittoolbox.com/> (accessed January 4, 2010)  
An online community in which peers share knowledge about information technology
- Microsoft Developer Network  
  
<http://msdn.microsoft.com> (accessed April 7, 2010)  
  
A library of programmer documentation for Visual Studio, SQL Server, and other Microsoft platforms
- Microsoft Project  
  
<http://www.microsoft.com/project/en/us/default.aspx> (accessed June 22, 2010)  
  
Contains product documentation and training presentations for Microsoft Project
- Microsoft Visio  
  
<http://office.microsoft.com/en-us/visio/FX100649211033.aspx?CTT=96&Origin=CL100636311033> (accessed June 22, 2010)  
  
Contains product documentation and training presentations for Microsoft Visio
- Sun Java Developer Network  
  
<http://java.sun.com> (accessed April 7, 2010)  
  
A library of programmer documentation for Java
- The Value of Project Lifecycle Methodology  
<http://www.lifecyclestep.com/open/401.0HomeValue.htm> (accessed January 4, 2010)

Site specializing in developing, consulting, and training in business methodologies

- Wideman Comparative Glossary of Project Management Terms  
<http://www.maxwideman.com/pmglossary/> (accessed January 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.

### Search by Key Terms

- Acceptance testing
- ActionScript
- Algorithms
- Application testing
- ASP.NET
- Bug tracking
- Cascading Style Sheets
- Class diagram
- Collaboration diagram
- CSS
- Data definition language
- Data manipulation language
- Database schema
- Debugging
- Flash
- Flowchart
- HTML
- Informal scenarios

- Integration testing
- Integration tests
- Interaction diagram
- Java
- JavaScript
- MySQL
- Object diagram
- Persistence
- PHP
- Project closure
- Project risks
- Project schedule
- Project scope
- Project teams
- Pseudocode
- Regression testing
- Relational database design
- Requirements gathering
- Requirements specification
- Requirements validation
- Risk assessment
- Risk management
- Sequence diagram
- SQL
- System integration
- Use case diagram
- Use case scenario

- Product design
- Risk assessment
- Ruby on Rails
- SQL Server
- Task network
- Unit testing
- Unit tests
- User documentation
- User interface design
- Visio
- Visual Basic
- WBS
- Work breakdown structure



## Course Evaluation and Grading

### Evaluation Criteria

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Team Project	
• Project Management Documentation	10%
• Teamwork	10%
• Presentation	10%
• System Design	20%
• System Implementation	35%
Individual Skills Assessment	15%
<b>Total</b>	<b>100%</b>

### 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)*