

**CSC305: Software Engineering  
Adult Degree Program  
Spring 2007**

**SYLLABUS**

**COURSE DESCRIPTION**

Concepts and skills to analyze, design, implement, and manage information systems using modern analysis and design methodologies. Significant real-world software development project covering all the phases of information systems development life cycle using high-level automated analysis and design tools. Experience with other important skills such as fact-finding, communications, and project management

**CREDITS : 5**

**OBJECTIVES**

This course fulfills suggested curriculum requirements of ACM, IEEE, and the Computer Science Accreditation Board for Computer Science departments. The course will provide the student with an overview of the tools and techniques of software engineering.

This course is designed to prepare the student to be able to join a team involved in software development projects and quickly become productive in that environment.

The course will provide the student with experience in system documentation by incorporating a major project that is to be completed by the students throughout the course of the semester. This project will involve significant writing, programming, and presentation/speaking as the students design, implement, and test their solutions. Each team will be expected to present its solution to the class providing the students an opportunity to prepare and deliver a technical presentation.

Object-Oriented development has become a hot topic in software development. This course explains object-oriented development by presenting the basic principles. It then introduces object-oriented analysis as a way of modeling user requirements and possible software solutions, and as a means of implementing the results, of offering possible software solutions, and as a means of implementing the results of object-oriented analysis. Transition from analysis to design is discussed along specific design techniques. Prototypical object-oriented programming languages are introduced and compared.

Upon completion of this course the student should be:

- describe a phased approach to information systems development and describe cross-life cycle activities that overlap the entire cycle.
- to present a technical material to a group.
- to create precise and informative documents for each stage of software development.
- to apply several approaches to software design such as data flow oriented, data structure oriented, and object oriented techniques.
- understand the importance of quality assurance and reliability of software systems.
- understand the process of graphic user interface(GUI) and dialogues design using Visual Basic.NET
- access relational data using ADO.NET
- apply object-oriented techniques to VB.NET programs

**Educational Objectives**

- Broaden existing software design and programming skills using current methodologies.
- Enhance existing software document skills
- Incorporate design-for-maintainability considerations through appropriate project work

**Professional Objectives**

- Exposure to design and programming technology that is widely-used in industrial and research environments.
- Enhance planning and time management skills by making students work through a significant project with multiple, staged deliverables.

- To provide a practical software development techniques that will prepare graduates to lead large software projects in industry.
- Define the systems analyst role and responsibilities in a typical organization.
- Prepare and use various information gathering techniques for eliciting user information requirements and system expectations.
- Develop a better plan for becoming a better systems professional or user/manager of systems, by understanding personal strengths and weaknesses and matching those with the ethical success factors of a modern business manager.
- Write a formal technical or professional report that organizes a systems documentation

### **COURSE OVERVIEW**

How to specify requirements using Use Cases

How systems are specified using Object, Interaction and Dynamic Models

How to build an object oriented solution based on a suitable target system architecture

How to specify, design, build and deliver a two-tier client/server and a three-tier application systems.

**Instructor** : Martin Dwomoh-Tweneboah

Office : Renshaw 208

Office Phone : 2426

E-mail : mdwomoh@linfield.edu

Office Hours : Daily 2.00 – 4.00 PM or by appointment.

**Text(Required)** : Systems Analysis and Design With UML 2.0 - An Object-Oriented Approach, Second Edition

**Bradley Millspaugh**; Programming in Visual Basic.NET, Irwin McGraw Hill

**Software**: Visio, Visual Studio.NET, Microsoft SQL, Rational Rose Enterprise.

### **CLASS AND GROUP DISCUSSION**

Active involvement of each student in class and group discussions is essential for a course of this kind. Students are expected to make pertinent and substantive contributions to every group discussion because team work plays important role in software development projects.

### **GRADING**

Your performance in this class will be measured by various assignments, quizzes, and the deliverables for the course project. Written assignment should be done neatly, typed, stapled in the upper left corner, so that it is : (a) easy to grade, and (b) useful as a study aid. Assignments submissions that do not meet minimal standards for acceptability in the workplace (completeness, neatness, readability, etc.) will be returned ungraded.

### **ASSIGNMENT SUBMISSION POLICY**

Late assignments will not be graded. One of the main objectives of this course is to learn how to deliver a project on time. Therefore, late assignments will not be tolerated. Assignments must be submitted on or before the deadline which will normally be the Saturday of our meeting weekend.

If you simply don't turn in assignments or deliverables for your project, your course grade will be calculated with the missed portion counting for 0 points.

**Grading Scale:**

|          |    |
|----------|----|
| 95 – 100 | A  |
| 90 – 94  | A- |
| 85 – 89  | B+ |
| 80 – 84  | B  |
| 75 – 79  | B- |
| 70 – 74  | C+ |
| 65 – 69  | C  |
| 60 – 64  | C- |
| 50 – 59  | D  |
| Below 50 | F  |

**COURSE MATERIALS**

Assignments, quizzes, and other relevant course materials will be posted at the course web page. It is your responsibility to check the web page for due dates on regular basis.

**ACADEMIC HONESTY**

Cheating and plagiarism will not be tolerated. Any student found to be engaging in either of these activities at any point in the course will receive a failing grade for the entire course and may be subject to further college sanctions.

**CLASSROOM AND GROUP DISCUSSIONS**

We must assume that we are all persons of intelligence and good will who are here to learn from each other in a team environment. Group discussions should not be a forum to impose our ideas on others. For the academic endeavor to succeed, we must treat each other with civility, courtesy and respect. Software development involves team work and all contributions by group members should be discussed and analyzed thoroughly.

**DIFFICULTIES**

If you find you are having problems with the class – the use of the software package, case tool, attendance, keeping up with the reading, fitting into a group, please let me know. I am always available to help you, but I have to know about the problem while it's going on and before the end of the course. The last few weeks to the end of the course is not the best time to ask for help.

**DISABILITY SUPPORT**

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should know of, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, no later than the first week of the term.

**CELL PHONE USAGE , EMAIL AND WEB BROWSING**

All cell phones should be turned off during lectures. Unless otherwise instructed, all applications, including browsers and email clients must be closed during lectures.

## COURSE OUTLINE

### Course Expectations

**Outcome :** After this session, student should understand the scope and requirements of this course as well as the knowledge and skills need to be mastered to succeed in this course

Introduction to the software tools to be used.

### Software Development Life Cycle and Software Management

**Outcome** After this session the student should be able to:

- identify the various phases of a development life cycle – requirements analysis, analysis phase, design phase, coding phase, testing phase.

### The vocabulary of Object Oriented Technology and The Unified Modeling Language Notations

**Outcome:** At the end of this session, students should be able to :

- understand the concepts and vocabulary of object technology
- visualize and identify the various notations used in the Unified Modeling language

### Requirements Gathering

**Outcome:** At the end of this session, students should be able to :

- Understand the techniques used in gathering information about the systems functionalities.

### System Behavior – Use Cases

Analysis of the needed system behavior from a use case approach

Development of scenarios for use cases

**Outcome :** At the end of the session, students should be able to

- capture system requirements using Use Cases.

### Finding Classes

Application of use case analysis to discover classes in the system

Definition of relationships needed for object interaction

Operations and Attributes

Definition of class structure and behavior

**Outcome:** At the end of the session, student should be able to :

- capture system requirements using object-oriented concepts, including classes, objects, attributes, operations, relationships and multiplicity

### Sequence and collaboration diagrams

Development of state transition diagrams to graphically show the behavior of an object

**Outcome :** At the end of the session, students should be able to:

- create sequence and Collaboration diagrams with a Visual Modeling tool
- create state diagrams with a Visual Modeling tool

### User interface design and Accessing the Database Using Visual Basic.NET

**Outcome:** At the end of the session, students should be able to:

- Apply appropriate user interface strategies to a system.
- Use the design models to plan and coordinate a user interface.

Select proper screen-based controls for input attributes that are to appear on a GUI input screen

### Programming with Microsoft ADO.NET

**Outcome:** At the end of the session, students should be able to:

- Build data-centric applications with Microsoft® ADO.NET, Microsoft® SQL Server™ 2000, and the Microsoft® .NET Framework.

### Testing and Deployment

**Outcome:** At the end of the session, students should be able to :

- understand the purpose and effectiveness of various testing strategies.