SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: SYSTEMS ANALYSIS AND DESIGN

Course No.: CET311

Program: COMPUTER ENGINEERING TECHNOLOGY

Semester: FIFTH (5)

Date: JANUARY 1994

Author: FRANK TURCO

Previous Outline Dated: JANUARY 1993

APPROVED: ______________ 94-01-11

Dean Date

COURSE OUTLINE

PREREQUISITES: The two year Technician Diploma or Permission granted by the Dean

LENGTH OF COURSE: 5 HOURS PER WEEK

TOTAL CREDIT HOURS: 96

I. PHILOSOPHY/GOALS

Many software systems are still being produced that are unreliable, over budget, poorly documented and not well suited to the user. A well engineered software system must be reliable, understandable, and maintainable.

A disciplined software development approach is absolutely critical to minimize the common problems with software. Most organizations follow a structured and disciplined approach to software development. They use different tools, techniques and methodologies and levels of sophistication but for the most part follow the System Development Life Cycle (SDLC).

In this course we will follow a systematic approach to systems analysis and design. The student will gain a thorough understanding of the SDLC through the preparation of deliverables (documents, discussions, coding) at each stage. We will also discuss and use some of the newer developments such as the modified SDLC, Rapid Application Design, Object Oriented Programming and others.

In all the tools and techniques, the most important component will always be communication. Therefore, communication is the key to success in software development and thus documentation, oral, written and interpersonal communication skills will be the main focus of this course.

II. STUDENT PERFORMANCE OBJECTIVES

Upon successful completion of this course the student will:

1. Understand why Systems Analysis is so important.
2. Appreciate the need for effective communication.
3. Understand the People aspects of Systems Development.
4. Use effective interviewing techniques.
5. Understand the Systems Development Lifecycle.
6. Use a variety of CASE tools and techniques to properly design systems such as:

   ENTITY RELATIONSHIP DIAGRAMS
   DATAFLOW DIAGRAMS
   COMMON DATA DICTIONARY

7. Use MICROSOFT PROJECT to do project planning.
8. Use EXCELERATOR as the repository of Dataflow Diagrams and Entity Relationship Diagrams.
9. Produce deliverables at the various stages of the Software Development Lifecycle such as:

   PROBLEM STATEMENT
   FEASIBILITY STUDY
   PROJECT PLAN
   REQUIREMENT SPECIFICATION DOCUMENT
   FUNCTIONAL SPECIFICATION DOCUMENT

III. TOPICS TO BE COVERED

This particular course assumes the student has already attained significant programming skills and techniques through previous courses. The course focuses on the crucial analytical side of software development. It takes much more than a great programmer to be successful in building software.

Here are the specific topics we will discuss and be assessed on:

1. The need for effective planning.
2. The need to manage the project.
3. The stages of the Software Development LifeCycle.
4. Software development methodologies such as:
   a) Joint Application Design
   b) Object Oriented Programming
   c) Rapid Application Design
   d) CASE
   e) Modified SDLC
   f) Reverse Engineering
   g) DDLC
5. The Human Element of Software Development
6. Proper Documentation Skills.
7. Walk Throughs and informal presentations.
8. The use of Microsoft Project for project planning and benchmark charting.
9. The use of Excelerator to produce the graphical representations of the various diagrams.

IV. LEARNING ACTIVITIES / REQUIRED RESOURCES

REQUIRED RESOURCES

TEXTBOOK:

1. **"Systems Analysis & Design Methods" by Whitten / Bentley / Barlow**

Additional reference material will either be given to the students or placed in the library for the student's use.

Instructor's Handouts, Guidance, and Material as it relates to the individual topics.

Individual and Group Assignments to be formally prepared as assigned.

IV. LEARNING ACTIVITIES / REQUIRED RESOURCES

In this course, the student will be introduced to a variety of tools, techniques and methodologies that will allow them to properly analyze and design computer solutions. The following blocks will not necessarily follow any chronological order. Topics will be covered as deemed appropriate. The main focus is to have the students appreciate that developing software is much more than just having the technical ability to program. Communication, people, technologies, environment and many other factors are also critical factors towards successful development.

BLOCK 1 INTRODUCTION TO SYSTEMS ANALYSIS AND DESIGN

LEARNING ACTIVITIES

At the end of this block, the student shall be able to:

1. Describe the historical evolution of Systems Analysis and Design.
2. Describe what systems analysis is.
3. Describe what systems design is.
4. Describe the quality, effectiveness, productivity and political aspects of Software Projects.
5. Define what a system is and what the different system categories are.

RESOURCES

TEXTBOOK

CHAPTER 1,
APPENDIX B, C

INSTRUCTORS
HANDOUTS

TRANSPARENCIES
CLASS NOTES

ARTICLES

BLOCK 2 PEOPLE ASPECTS IN SOFTWARE DEVELOPMENT

LEARNING ACTIVITIES

At the end of this block, the student shall be able to:

1. Define and describe the categories of people involved in software development.
2. Define and describe the three main categories of users as well as the different objectives they have.
3. Describe the role of a systems analyst in a systems development project.
4. Describe managements role in a systems development project.
5. Describe the roles of others in a software project.

RESOURCES

TEXTBOOK

CHAPTER 2, 3

INSTRUCTORS HANDOUTS

TRANSPARENCIES

ARTICLES

BLOCK 3 THE SYSTEMS DEVELOPMENT LIFE CYCLE METHODOLOGY

LEARNING ACTIVITIES

At the end of this block, the student shall be able to:

1. Describe the concept of a project life cycle.
2. Describe the characteristics of the classical project life cycle.
3. Define and describe the different components of the systems development life cycle.
4. Describe the differences between radical and conservative life cycles.
5. Describe the prototyping approach.
6. Explain the changes that have taken place in structured analysis.
7. Describe why automated tools are important to the future of systems analysis.

RESOURCES

TEXTBOOK

CHAPTER 4, 5

INSTRUCTORS HANDOUTS

TRANSPARENCIES

ARTICLES

RESEARCH MATERIAL

BLOCK 4 PLANNING A SOFTWARE PROJECT (PROJECT MANAGEMENT)

LEARNING ACTIVITIES

This section will be covered in greater detail in the Software Engineering course. However, it is essential to be exposed to the approach and concepts of project management at this stage. The follow up course will develop these skills in greater detail.

At the end of this block, the student shall be able to:

1. Understand the concept of planning and its relevance.
2. Define project goals and requirements.
3. Discuss the relationship of planning with respect to project size.
4. Discuss the project planning development process including:
   a) Project Phases
   b) Milestones, Documents, Reviews
   c) The cost aspects of each phase of the project
   d) Prototyping
   e) Successive versions
5. Discuss the project planning organizational structures including:
   a) Project Format
   b) Project team structure
   c) Project quality assurance
   d) Project verification and validation
6. Produce the project feasibility study. (also known as engineering study)

RESOURCES

TEXTBOOK
CHAPTER 6,12

APPENDIX A, D

INSTRUCTORS
HANDOUTS

TRANSPARENCIES

MICROSOFT
PROJECT

RESEARCH
ARTICLES

SYSTEMS ANALYSIS AND DESIGN CET311

BLOCK 5 MODELLING TOOLS AND SOFTWARE DELIVERABLES

LEARNING ACTIVITIES **RESOURCES**

This section will represent the majority of the material covered in this course. There will be extensive use of the tools and techniques to properly analyze and design computer systems. We will endeavour to use these tools and techniques throughout the course. The student will also be exposed to the software deliverables in this course. The follow up course will develop these skills in greater detail. Some of these tools and deliverables will be covered in greater detail than others.

At the end of this block, the student shall be able to:

1. Define, describe and use the major modelling tools such as:
   a) Dataflow Diagrams
   b) Data Dictionary
   c) Process Specifications
   d) Entity-Relationship Diagrams
   e) State-Transition Diagrams
   f) Context Diagrams
   g) Joint Application Design Sessions (JAD)
2. Produce software deliverables at each stage of the SDLC such as:
   a) Problem Statement
   b) Feasibility Study
   c) Project Plan
   d) Requirement Specification Document
   e) Functional Specification Document
   f) Computer System Design Document
   g) Managerial Presentations
3. Use Excelerator for ERD's & DFD's.

TEXTBOOK CHAPTER 7, 8, 9, 10, 11

INSTRUCTORS HANDOUTS

TRANSPARENCIES

MICROSOFT PROJECT

RESEARCH ARTICLES

EXCELERATOR

SYSTEMS ANALYSIS AND DESIGN

LEARNING ACTIVITIES SUMMARY

We will use a variety of tools and techniques to accomplish the objectives as set out in a previous section.

The following are some of the major activities that will take place in this course.

Students will be asked to formally and informally answer several specific questions on an individual basis. These questions will reinforce the topics and concepts as discussed in the text material and the lectures.

Students are also required to be team players and work in small groups to answer some of the questions and solve mini cases. The objective here is to build a solid team atmosphere as well as having students appreciate that there isn't always a clear cut answer to development and people with different perspectives can improve the results. The students are to bring motivation, participation and good listening skills to the table to help each other come up with a better collective solution.

Once we work with the variety of tools used in SDLC, students will be required to provide graphical documentation such as DataFlow Diagrams and Entity Relationship Diagrams. If the facilities are available, the students will be required to use productivity tools such as EXCELERATOR to produce this documentation. We will also use tools for project management such as Microsoft Project. We will also simulate a common repository (dictionary) for software development and maintain internal communication documentation either on the VAX or the LAN or a single protected Hard Drive.

VI. SPECIAL NOTES

1. Students with special needs are encouraged to discuss required accommodations confidentially with the instructor.
2. Your instructor reserves the right to modify the course as he/she deems necessary to meet the needs of students.

VII. ASSESSMENT

Tests and Quizzes	60%
Assignments and Lab Work	40%

The tentative breakdown is as follows:

2	Formal Theory Tests	at 15 %	each
2	TAKE HOME Tests	at 15 %	each
4	Assignments	at 5 %	each
2	Assignments	at 10 %	each

Some minor modifications to the above percentages may be necessary. The instructor reserves the right to adjust the mark up or down 5% based on attendance, participation, leadership, creativity and whether there is an improving trend.

* All Assignments must be completed satisfactorily to complete the course. Late hand in penalties will be 5% per day. Assignments will not be accepted past one week late unless there are extenuating and legitimate circumstances.

* Due to the heavy emphasis on group effort and team work, late hand ins will not be allowed for some assignments. Absenteeism and lack of group cohesiveness will disrupt all members of the group and will not be tolerated.

* The instructor reserves the right to adjust the number of tests, practical tests and quizzes based on unforseen curcumstances. The students wil be given sufficient notice to any changes and the reasons thereof.

VII. ASSESSMENT (CONTINUED)

GRADING SCHEME

1. TESTS

Written tests will be conducted as deemed necessary; generally at the end of each block of work. They will be announced about one week in advance. Quizzes may be conducted without advance warning.

2. ASSIGNMENTS

Assignments not completed by the assigned due-date will be penalized by 5% per day late. All assignments must be completed satisfactorily to complete the course.

3. GRADING SCHEME

A+	90 - 100%	Outstanding achievement
A	80 - 89%	Excellent achievement
B	70 - 79%	Average Achievement
C	55 - 69%	Satisfactory Achievement

U Incomplete:Course work not complete at Mid-term. Only used at mid-term.

R Repeat

X A temporary grade that is limited to instances where special circumstances have prevented the student from completing objectives by the end of the semester. An X grade must be authorized by the Chairman. It reverts to an R if not upgraded in an agreed-upon time, less than 120 days.

SYSTEMS ANALYSIS AND DESIGN

4. UPGRADING OF INCOMPLETE

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when the student's performance warrants it. Attendance and assignment completion will have a bearing on whether upgrading will be allowed. A failing grade on all tests will remove the option of any upgrading and an R grade will result. The highest grade on re-written tests or assignments will be 56%.

Where a student's overall performance has been consistently unsatisfactory, an R grade may be assigned without the option of make-up work.

The method of upgrading is at the discretion of the teacher and may consist of one or more of the following options: assigned make-up work, re-doing assignments, re-writing of tests, or writing a comprehensive supplemental examination.