SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: **SYSTEMS ANALYSIS AND DESIGN**

Code No.: CSD202

Program: **COMPUTER PROGRAMMER/ANALYST**

Semester: Three

Date: **FALL 1997**

Author: **Willem de Bruyne**

Date: **May 1997** Previous Outline Dated: **Sept 1996**

APPROVED: ______________________ May 12/97

Dean Date

TOTAL CREDITS: 5

PREREQUISTES: NONE

LENGTH OF COURSE: 4 hours per week

2 - 1 hour theory classes
1 - 2 hour lab

I. Course Description:

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 Prototyping, Rapid Application Design.

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.

The student will use the CASE tool Excelerator to assist them in planning, analysis, design, documentation and construction of a computer based information system in accordance to one or more software development methodologies.

II. TOPICS TO BE COVERED

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.

SPECIFIC TOPICS

1. Introduction to System Analysis and Design
2. People aspects in software
3. Systems Development Life Cycle
4. Project Planning and Management
5. Modelling Tools and Software

III. 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.

LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

A. Introduction to System Analysis and Design

Learning Outcomes:

At the end of this block, the student will demonstrate the ability to:

- Describe the historical evolution of Systems Analysis and Design.
- Describe what systems analysis is.
- Describe what systems design is.
- Describe the quality, effectiveness, productivity and

political aspects of Software Projects.

- Define what a system is and what the different system categories are.

LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

B. People aspects in software

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

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

LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

C. Systems Development Life Cycle

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

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

LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

D. Project Planning and Management

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

- Understand the concept of planning and its relevance.
- Define project goals and requirements.
- Discuss the relationship of planning with respect to project size.
- 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
- Discuss the project planning organizational structures including:
    a) Project Format
    b) Project team structure
    c) Project quality assurance
    d) Project verification and validation
- Produce the project feasibility study. (also known as engineering study)

LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

E. Modelling Tools and Software

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:

○ understanding Excelerator/IS terminology
○ learn how to log onto Excelerator
○ know how to change Excelerator project defaults
○ understanding effective use of the graphics profile

- know how to use the action keypad options
- learn how to print graphics drawings
- know how to transfer printed output to a file
- understand how to set print options
- know how to save project work using the Backup feature
- know how to transfer backup files into the project
- understand how to use the Export feature
- understand how to Import project backup
- know how to exit Excelerator and the Exit options
- learn how to move objects, connections, text
- know how to copy objects
- know how to delete objects and other graph components
- understand the screen refresh option
- learn how to centre the drawing
- know how to add a line or a block of text to the drawing
- learn how to add lines and boxes to the drawing
- understand how to disconnect an object from its XLDictionary
- know how to create and delete a drawing title
- learn how to create data flow diagrams
- understand the various methods of connecting objects
- know how to create labels and data dictionary descriptions
- understand how to link data flow diagram components to detailed drawings and descriptions
- learn how to create child data flow diagrams for processes balancing
- know how to create a presentation graph
- learn how to describe objects on the graph
- understand how to link an object to the XLDictionary
- understand how to create an explosion path
- learn how to create data dictionary record structures
- learn how to print data dictionary entries
- understand how to delete, copy, and rename data dictionary entries

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.