NEW ZEALAND INSTITUTES OF TECHNOLOGY AND POLYTECHNIC QUALIFICATIONS IN INFORMATION & COMMUNICATIONS TECHNOLOGY

PRESCRIPTION: OO600 OBJECT ORIENTED ANALYSIS AND DESIGN

AIM OF MODULE:	The student will understand and apply analysis and design techniques using the Unified Modelling Language.
CREDITS:	7
KNOWLEDGE ASSUMED FROM:	OO500
STUDENT LEARNING HOURS:	70
CONTENT REVISED:	2004
PRESCRIPTION EXPIRY DATE:	Nov 2011
NOTE:	It is expected students would study this concurrently with or subsequent to an object-oriented programming language module at the 600 level.

Level and Assessment Schedule

TOPICS	Highest Skill Level R	C	A	P	Suggested Assessment Percentage
1. Purpose of diagrams	*				10
2. Relationship between diagrams		*			10
3. Development process		*			10
4. Model creation				*	70
					100

LEARNING OUTCOMES

The student will:

C	1	Understand and describe the syntax and purpose of each UML diagram.
A	2	Understand and demonstrate the relationship between each of the UML diagrams.
A	3	Understand and demonstrate the iterative, incremental and evolutionary nature of analysis and design in model development using UML and how such development is controlled.
P	4	Create a suitable set of UML models for a given case study.

CONTENT

1 PURPOSE OF DIAGRAMS

- All diagrams of the UML are described and the purpose of each explained.

2 RELATIONSHIP BETWEEN DIAGRAMS

- How diagrams are related and the typical order in which they are created is demonstrated by solving a software problem.

3 PROCESS

- The meaning of iterative, incremental and evolutionary development and the level of detail required of models created during the analysis and design phases are described.
- How such development impacts on project planning and control is described, and how such development is managed is explained.
- A suitable process model (development methodology) is described and used in solving a software problem (see note 4).

4 MODEL CREATION

- From a given case study a complete set of analysis and design documents are developed using object-oriented techniques and an appropriate development methodology.

- A CASE tool is used to develop the models and to generate class header code if possible (see note 3).

- Suitable models include:
  - Use case diagram
  - Class diagram
  - Interaction (sequence or collaboration) diagram for selected objects
  - Statechart for selected objects
  - Activity diagram for workflows or selected methods
  - Component diagram
  - Deployment diagram

- Diagrams may be provided at both the analysis level and design level of detail.

NOTES

1. To gain full benefit from this module it should be taught concurrently with a 600 (200) level object-orientated programming language. Case studies and assessments could then be amalgamated to provide a complete picture of the analysis, design and implementation and testing of an object-oriented system. For example: providing designs for a programming exercise, re-engineering designs from a programming solution, using Use Case models to develop test plans.

2. UML standard version 1.1 or above is recommended.

3. Suitable CASE tools include Rational Rose, Visible Analyst, System Architect.

4. Suitable process models (development methodologies) include OMT, Objectory, OPEN, Rational Unified Process.

LEARNING RESOURCES

Suggested textbooks:

- Bennett, S., McRobb, S. & Farmer, R. (1999). *Object-Oriented Systems Analysis and Design and Design using UML.* England: McGraw-Hill.

- Shelly, G.B., Cashman, T.J., & Rosenblatt, H.J. (2001). *Systems Analysis and Design*. Washington: Course Technology.

TEACHING RESOURCES

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modelling Language Reference Manual*. Massachusetts: Addison Wesley Longman.