2910 Object-Oriented Analysis and Design

Introduction

2910 Object-Oriented Analysis (OO) and Design provides a comprehensive treatment of what has now become the preferred approach to the analysis and design of large-scale software systems. This subject describes in detail processes and related workflows as well as people and artefacts involved in the analysis and design of Information Systems (IS). Domain modelling and analysis modelling are discussed in conjunction with the OO paradigm and the industry-standard Unified Modelling Language (UML). The focus will be on building explicit and traceable IS models, and extensible and reusable IS architecture.

Who should attend

- Architects and software designers involved in the development of software systems
- IT developers undertaking analysis and design activities
- Those involved in software development projects that are deploying an OO approach
- Managers looking to adopt OO development practices within their organisation

Learning objectives

Upon completion of this subject, students should be able to

- explain the fundamental concepts behind the OO perspective
- analyse problems through the development of structural, behavioural and state-chart models
- use the RUP development methodology to systematically plan and conduct OO development projects
- capture user requirements through use-case modelling and domain object modelling
- examine use-cases to analyse and build an understanding of the problem
- develop design models that reflect OO solutions
- transfer best OO practices into projects in the workplace

Delivery method

The subject is delivered online over a 12-week period, with an assigned Professor acting as mentor. The class will comprise students from different countries and industry backgrounds. Practical assignments and discussions help to stimulate learning and knowledge exchange, while an examination at the end of the subject will help students review and apply the knowledge and skills learnt.

Prerequisites

Students are recommended to have project experience of systems development work.

Assessment	
Assignments (team and individual)	45%
Discussion board activities	30%
Final examination	25%

Syllabus

Segment 1: Introduction

Students are introduced to the syllabus, the resources and communication tools available within the course.

Segment 2: The Object Perspective

The segment lays the foundations of the Object-Oriented (OO) approach. The concept of objects, and how they differ from classes, is introduced. The fundamental principles upon which the OO approach is based are explained in detail. This includes principles such as encapsulation, aggregation, inheritance and polymorphism. The segment also introduces the Unified Modelling Language (UML), the de-facto industry standard for general-purpose OO modelling proposed by the Object Management Group.

Segment 3: Object Diagramming and Notational Techniques

The segment explains the diagramming and notational techniques within UML which are used to describe and represent OO systems. The visual nature of UML and how it is supported by Computer Aided Software Engineering (CASE) tools are described. The creation of structural models in UML, through the use of class diagrams, component diagrams and deployment diagrams, is illustrated. The creation of behavioural models in UML, through the use of use-case models, state-chart diagrams and activity diagrams, is also explained.

Segment 4: Development Methodology

The segment emphasises the importance of development methodology as a means of systematically applying OO processes in the context of a development life cycle. Software engineering best practices are touched upon. This is followed by an overview of the Rational Unified Process (RUP), one of the most popular and widely-used development frameworks. RUP is an iterative framework that splits the project life cycle into four phases, namely inception, elaboration, construction and transition. The segment provides treatment of the workflows and models within each phase.

Segment 5: Requirements Modelling

The segment provides a detailed treatment of requirements modelling. Use-case models are presented as a means by which designers can understand user requirements. Use-cases represent how users envisage a system being used and the main actors involved. This is followed by an explanation of domain object models and how such models can be used to capture objects in the problem domain. The segment provides further guidance on documenting the requirements model.

Segment 6: Analysis Modelling

The segment provides a more in-depth treatment of analysis modelling. Use-case analysis is presented as a way of identifying analysis objects. The segment also discusses the definition of analysis classes and the allocation of behaviour to classes. Guidance is provided on how analysis models should be documented, paying particular attention to analysis model workflow.

Segment 7: Design Modelling

The segment describes important aspects of design modelling. The difference between design and analysis is highlighted. The concept of persistent objects is introduced as a basis for designing persistence in a system. The segment explains how systems can be partitioned into subsystems each dealing with a particular aspect of the system. Attention is also given to the design of classes, and important design characteristics such as cohesion and coupling. Guidance is also given on documenting the design model.

Segment 8: Advanced Principles, Strategies and Best Practices

The segment looks at some of the advanced principles and strategies for Object-Oriented analysis and design. These include encapsulation and connascence, encumbrance and cohesion, and type conformance and closed behaviour. The segment also examines a number of best practices, such as the application of analysis and design patterns.

Required textbook

None

Global Faculty

Subject Authors

U21Global subjects are created by acknowledged experts in their field, usually senior academics who have strong understanding of postgraduate requirements. The subject content is further reviewed by academic specialists who appraise the subject from an independent perspective, ensuring a high-quality, professional product.

2910 OBJECT-ORIENTED ANALYSIS AND DESIGN was jointly developed by U21Global and the Institute of Systems Science (ISS), National University of Singapore. ISS specialises in providing professional information technology continuing education to managers and IT practitioners. NUS ranks as one of the top global universities in Asia and Australia.

Professors

Students' progress will be guided by dedicated Professor Facilitators based around the world. They provide an international perspective and impart knowledge through a wealth of experience in their field of specialisation. Our Professor Facilitators will help students make sense of the information to enable students to transform the information into knowledge and creative solutions.

Adam HUARNG

Adam Huarng is Professor of Information Systems at California State University, Los Angeles, US. He previously taught at Minnesota State University and Purdue University, Fort Wayne, where he also served as Chair for the Computer Science Department. Dr Huarng's research interests are in the areas of software development, database analysis and eCommerce development. He has also worked as a software engineer for the Telecommunication Labs in Taiwan, Federal Express in Memphis, Tennessee and Micro Tech Concepts in Los Angeles, California. He received his PhD in Business Administration with concentration in Management Information Systems from The University of Memphis.

France CHEONG

France Cheong is Senior Lecturer in the School of Business Information Technology at RMIT University in Melbourne, Australia. A former lecturer in the School of Computing Science at Queensland University of Technology, Dr Cheong was a Management Information Systems Consultant in Mauritius, where he originates. His research interests lie mainly in the areas of modelling and simulation of business and other systems, such as eBusiness, mobile commerce, economics and finance, management and manufacturing systems. He received his PhD in Computer Systems Engineering and Master of Computer Science from La Trobe University, Australia.