Module: Software Analysis & Design 361

Module name:	Software Analysis & Design361
Code:	SWA361
NQF level:	6
Type:	Speciality – Diploma in Information Technology (Programming)
Contact Time:	34 hours
Structured time:	6 hours
Self-directed time:	40 hours
Notional hours:	80 hours
Credits:	8
Prerequisites:	SWA261

Purpose

This module covers the method and methodologies that are required for eliciting business requirements and transforming the requirements into software implementation. It covers a systematic methodology for analysing a business problem or need, determining what role computer-based technologies play in addressing the business need, and specifying alternative approaches to acquiring the technology capabilities needed to address the business requirements and the requirements for the information systems solution.

Outcomes

Upon successful completion of this module, the student will be able to:

- Understand a range of methods of enquiry in the discipline and their suitability for application for the investigation of business needs within the agile framework, writing clear, concise user stories, documenting and converting them into technical specifications.
- Analyse and model requirements using UML notations for the realization of specific use cases.
- Demonstrate and communicate their understanding of object-oriented systems analysis and design
- Take decisions and articulate ethical, cultural, and legal issues and their feasibilities among alternative solutions.
- Develop software methods from a detailed analysis of design artefacts.

Assessment

- Continuous evaluation of theoretical work through written assignment, a formative, and a summative test.
- Continuous evaluation of project work, where the student must design, manage and report on the evaluation of testing methodologies and the selection of an appropriate methodology for a given scenario, justifying the choice made with well-formed arguments and evidence.
- Final assessment through a written examination.
- The assignments or projects collectively will count 30% of your class mark.
- All tests will collectively account for 70% of your class mark.

- Your class mark contributes 30% towards your final mark for the subject, while the final assessment accounts for 70% of your final mark.

Teaching and Learning

Learning materials

Prescribed books (EBSCO)

- 🕮 ***Kenneth Barclay and John Savage (2004)*** **Object-Oriented Design with UML and Java*. Oxford: Butterworth-Heinemann.***
- 🕮 ***Ambler, S. W. (2003)*** **The Elements of UML Style*. Cambridge, UK: Cambridge University Press.***
- 🕮 ***Satzinger, J.W., Jackson, R.B. and Burd, S.D. (2015).*** **Systems analysis and design in a changing world*. Cengage learning.***

Learning activities

The teaching approach consists of a combination of formal lectures on theoretical and practical concepts, solving real-world problems through exercises, demonstrations of feasible solution in a specific context and discussions of high-level design specifications. It is dialogue-oriented with a practical approach with One mandatory assignment and one project that must be completed during the module.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		27.0		13.0
Formative feedback		3.5		
Project	1	3.5		9.0
Assignment	1			3.0
Test	2		4.0	8.0
Exam	1		2.0	7.0
		34.0	**6.0**	**40.0**

Syllabus

- Agile Methodologies
- Object orientation core concepts
- Use-Case realization
- System Planning
- Requirement elicitation
- Requirement Analysis
- UML Modelling