



COLLEGE OF INFORMATION TECHNOLOGY
DEPARTMENT OF IT
COURSE SYLLABUS/ SPECIFICATION

Course Code & Title:	ITCS 514- Object Oriented Software Engineering
Weight:	(3-0-3)
Prerequisite:	None
NQF Level Allocated:	9
NQF Notional Hours / Credits:	120 notional hours/ 12 NQF credit

Description: This course focuses on object-oriented approach necessary to solve advanced and complex real-world problems. It is to understand a range of specialized theories, principles and concepts of object-orientation; object oriented software development process; the use of object-oriented design tools such as UML for modeling problem solutions. Topics include: Problem analysis and specification of software requirements; object-oriented design; reusability and design patterns; unit testing; advanced software development methodology such as Adaptive Object-Oriented Software Development.

Objective:

1. To critically explore various software development approaches and models.
2. To understand a range of specialized theories, principles and concepts of object-orientation, and their advantages.
3. To explain the development process of object oriented software.
4. To apply a range of specialist models and methods in developing software systems.
5. To research current trends in object oriented methodology and software engineering.

Semester: 1st 2019-2020

Instructor (s): Dr. Sohail Safdar

Office Telephone: EXT: 8698

Email (s): ssafdar@ahlia.edu.bh

Intended Learning Outcomes (ILOs):

A. Knowledge and Understanding		NQF Descriptor/ Level
A1	Concepts and Theories: Demonstrate critical understanding of a range of specialized theories, principles and concepts of software engineering and object oriented methodology.	Knowledge: theoretical understanding [Level 9]
A2	Contemporary Trends, Problems and Research: Demonstrate critical awareness of the current research, trends and advancements in software engineering and object oriented methodology.	Knowledge: theoretical understanding [Level 9]

B. Subject-specific Skills		NQF Descriptor/ Level
B1	Problem Solving: Use a variety of advanced approaches to identify problems, elicit, analyze, formulate and specify requirements of stakeholders, design and model solutions.	Knowledge: Practical Application [Level 9] Skills: Communication, ICT & Numeracy [Level 9]
B2	Modeling and Design: Apply a range of specialist models to model different perspectives of software using UML, and to apply different approaches to design software architecture and components.	Knowledge: Practical Application [Level 9]
B3	Application of Methods and Tools: Apply appropriate advanced methods and tools to software development process.	Knowledge: Practical Application [Level 9] Skills: Communication, ICT & Numeracy [Level 9]

C. Critical-Thinking Skills		NQF Descriptor/ Level
C1	Analytic skills: Conduct critical analysis of complex real world problems in order to identify possible solutions.	Generic Problem Solving & Analytical skills [Level 9]
C2	Synthetic: Critically review, and extend knowledge/skills/practices in software development to construct and reuse the multiple components of software/system.	Generic Problem Solving & Analytical skills [Level 9]
C3	Creative: Demonstrate creativity both in the application of knowledge and in the development of effective and efficient software applications to real world problems.	Generic Problem Solving & Analytical skills [Level 9]

D. General and Transferable Skills (other skills relevant to employability and personal development)		NQF Descriptor/ Level
D1	Communication: Develop an ability to effectively communicate with peers, and senior colleagues. Honing presentations skills in project report writing and oral presentation.	Communication, ICT and Numeracy Skills [Level 9]
D2	Teamwork and Leadership: Work effectively as a member of a study group, and where appropriate lead or take the initiative, to complete the group project/assignments according to deadlines.	Competence: Autonomy, Responsibility and Context [Level 9]
D3	Organizational and Developmental Skills: Learn to manage learning tasks independently and professionally with a view to inculcating skills for self-development and life-long learning in order to effectively prioritize, plan, manage and allocate appropriate resources to implement tasks in projects.	Competence: Autonomy, Responsibility and Context [Level 9]

Course Structure (Outline)

Week	Hours		ILOs	Topics	Teaching Method	Assessment Method
	Lec.	Lab				
1	3	-	A1	Course overview, software development process, software development approaches	Lecture/ Class Discussion	Major Exam (9th week) and Final Exam (Last Week)
2	3	-	A1	Principles of object orientation and its advantages, differences between an object-oriented approach and a procedural approach	Lecture/ Class Discussion	Major Exam (9th week) and Final Exam (Last Week)
3	3	-	A1, A2, D1, D3	Process models	Lecture/ Independent Learning	Major Exam (9th week) and Final Exam (Last Week)/ Assignment1 (literature reviews)
4	3	-	A1	Modeling Concepts, Introduction to UML	Lecture/ Class Discussion	Major Exam (9th week) and Final Exam (Last Week)
5	3	-	B1, C1	Object oriented software requirement analysis	Lecture/ In class exercises/ Case Studies	Project (week 15)/ Major Exam (9th week) and Final Exam (Last Week)
6	3	-	B1, B2, B3	Requirement specification and Use Case model	Lecture/ In class exercises/ Case Studies	Exercises/ Project (week 15)/ Major Exam (9th week) and Final Exam (Last Week)
7	3	-	B2, C1, C2	Software Architecture design	Lecture/ Case Studies	Major Exam (9th week) and Final Exam (Last Week)/ Project (week 15)

8-9	3	-	B2, C1, C2	Object oriented software design	Lecture/ Class Discussion	Project(week 15)/ Major Exam (9th week) and Final Exam (Last Week)
10-11	9	-	B2, B3, C2	Static and dynamic design modeling (Statechart model)	Lecture/ Case Studies/ In class exercises	Exercises/ Project (week 15)/ Major Exam (9th week) and Final Exam (Last Week)
12	3	-	A2, C2, B2, B3, D1, D3	Reusability and design patterns	Lecture/ Independent Learning	Exercises/Project (week 15)/ Final Exam (Last Week)
13	3	-	B3	Unit testing	Lecture/ tools demonstration	Project (week 15)/ Final Exam (Last Week)
14	3	-	A2, B2, D1, D3	Aspect-Oriented Software Development	Lecture/ Independent Learning	Assignment2/ Final Exam (Last Week)
15	3	-	A2, B1,B2, B3, C1, C2, C3, D1, D2, D3	Student Team Project Presentations	Regulations and Procedures for the project	Student Project Presentation (A2, B1,B2, B3, C1, C2, C3, D1, D2, D3)
16	2					Final Exam (A1,A2, B1,B2,B3, C1,C2)

* Formative assessment

Teaching Materials:

<p>Textbook(s):</p>	<ol style="list-style-type: none"> 1. Alan Dennis, Barbara Haley Wixom, David Tegarden, systems analysis and design: an object oriented approach with uml, 5th Edition, John Wiley & Sons, 2016. ISBN: 1-118-80467-8, 978-1-118-80467-4, 1-119-03026-9, 9781119030263. 2. Additional: Bernd Bruegge, Allen H.Dutoit, object-oriented software engineering using UML, Patterns, and Java ,3rd edition,Pearson,2014
<p>Handout(s):</p>	<p>Lecture Notes and Case Studies</p>
<p>Reference(s):</p>	<ol style="list-style-type: none"> 1. Ian Sommer Sommerville. <i>Software Engineering</i>, 9th Edition, Addison Wesley, 2010. 2. Simon Bennett, Steve McRobb and Ray Frammer, <i>Object-Oriented System Analysis and Design using UML</i>, 4th Edition, McGraw Hill, 2011. 3. Kim Waldén and Jean-Marc Nerson, <i>Seamless Object-Oriented Software Architecture: Analysis and Design of Reliable Systems</i>, 4th Edition, 2012. 4. Roger S Pressman. <i>Software Engineering – A Practitioners Approach</i>, McGraw Hill, 2006. 5. Michael Blaha and James Rumbaugh. <i>Object-oriented modeling and design with UML</i>, 2ed edition, Prentice Hall, 2005. 6. G Booch, J Rumbaugh, and I Jacobson. <i>The Unified Modelling Language User Guide</i>, 2nd Ed., Addison Wesley, 2005. 7. T. Lethbridge and R. Laganriere.<i>Object-Oriented Software Engineering: Practical Software Development using UML and Java</i>, 2nd Ed., 2004. 8. Alan Shalloway, James R. Trott.<i>Design Patterns Explained: A New Perspective on Object-Oriented Design</i>, Addison Wesley, 2001. 9. Karl J. Lieberherr. <i>Adaptive Object-oriented Software: The Demeter Method with Propagation Patterns</i>, PWS Publishing Company, 1996. 10. https://decibel.ni.com/content/docs/DOC-2875. <p>Journal papers will be available in the course website.</p>

Assessment

Method of Assessment	Description	Learning Outcomes	Weighting
Major Exam	The major exam will be an in-class 90 minutes exam that will consist of short-answer, essay, and problem solving questions, and cover the topics studied in the first 9 weeks.	A1,A2,B1,B2, B3,C1,C2	20%
Two Assignments	The assignments consist of essay, problem-solving and literature review. The purpose of the assignments is to assess students where they have to demonstrate their extensive and detailed knowledge and critical awareness of current research, trends and advancements in software engineering and object-oriented software development methodology. The assignments will also assess students' skills in applying problem solving approaches to elicit and analyze requirements of stakeholders, design and model solutions. Soft copy submission is required through the course page in Moodle where answers will be checked by Turnitin against plagiarism.	A2, C2, B2, B3, D1, D3	(10% + 10% =) 20%
Exercises	In-class exercises consist of problem solving and application of different tools and models to model and design software components. Each exercise will be one hour duration.	B1,B2, B3	Formative Assessment
Team Project	Student teams will propose and develop a medium-sized project of their own. Soft copy submission is required through the course page in Moodle where the project report will be checked by Turnitin against plagiarism.	A2, B1,B2, B3, C1, C2, C3, D1, D2, D3	20% Method of Evaluation: Completeness(8 marks) Quality(3 marks) Creativity(2 marks) Presentation: report quality, and oral presentation (flow of information, student personality, understanding and Q/A) (5 marks)

			Team Work (team members meeting minutes) (2 marks)
Final Exam	The final exam is comprehensive and will be of two hours duration. It will consist of short-answer, essay and problem-solving questions.	A1,A2,B1,B2, B3,C1,C2	40%
Overall:			100 %

Admissions	
Pre-requisites	None
Minimum number of students	4
Maximum number of students	25

Ahlia University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.ahlia.edu.bh/integrity for more information).