Teaching Plan

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SOFTWARE REQUIREMENTS AND DESIGN

BITP 2223	SEMESTER 2	SESI 2023/2024

BITP 2213 SOFTWARE ENGINEERING [3, 2, 2]

TYPE OF COURSE: C

EDITION: 1

UPDATED: 06-03-2024

1.0 LEARNING OUTCOMES

At the end of the lesson, students should be able to:

i. Analyze software requirement and design the software using object oriented approach (C4).

ii. Model software analysis and software design using object oriented approach (P3).

iii. Write formal software requirements specification document and software design document (A2, CTPS2).

2.0 SYNOPSIS

This course introduces the students to the Object Oriented Analysis and Design (OOAD) for software project development. This course covers the knowledge and skills to perform requirements elicitation, requrements analysis and negotiation, requirements specification, requirements validation, high level design and detailed design. The students will be taught to know sources of requirement, type of requirements, major activities in requirement engineering and writing Software Requirements Specifications (SRS). In design phase, the students will be exposed to designing software architecture, high level and detailed design which will be realized through refining the analysis models and producing Software Design Document (SDD).

3.0 PRE-REQUISITE

None

4.0 PRACTICAL

The students will form a group and propose a software project. Each group will exercise object oriented analysis and design (OOAD) to the proposed software project. 70% of the project effort is done in the lab. This is to ensure that all team members in each group is contributing to the project and also to avoid plagiarism.

5.0 REFERENCES

[1] A. Laplante, Phillip, Requirements Engineering for Software and Systems (Applied Software Engineering Series), 3rd Edition, Auerbach Publications, 2017

[2] Kenneth E. Kendall, Julie E. Kendall, Systems Analysis and Design, 9th Edition, Pearson 2014

[3] Chris Britton, Designing the Requirements: Building Applications that the User Wants and Needs, Pearson Education 2015

[4] Gerardus Blokdyk, Object-Oriented Analysis and Design a Complete Guide, Emereo Pty Limited, 2019

[5] Jim Arlow, Ila Neustadt UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2/E, Addison-Wesley Professional, 2012

6.0 COURSE IMPLEMENTATION

i. Lecture

- 2 hours per week for 14 weeks (Total = 28 hours)

ii. Laboratory Activities

- 2 hours per week for 14 weeks (Total = 28 hour)

7.0 COURSE EVALUATION

Assessment Method	CLO1 (C4)	CLO2 (P3)	CLO3 (A2, CTPS2)	Scheme, Rubric/ guideline
Quiz (1) = 10%		Q (10%)		RubricMP1.docx

Assignment (2) = 20%			A1 (10%) A2 (10%)	SchemaMT.docx
Mini Project (1) = 25%		P (15%)	P (10%)	RubricA1.docx
Mid Term (1) = 15%	MT (15%)			SchemaLT.docx
Final (1) = 30%	F (30%)			SchemaPA.docx
Total = 100%	**45%**	**25%**	**30%**	

8.0 STUDENT LEARNING TIME (SLT)

		Guided Learning Time				Independent Learning								Assessment Time				SLT
Week	**CLO**	**L**	**T**	**P**	**O**	**L**	**T**	**P**	**O**	**F**	**T**	**A**	**O**	**F**	**T**	**A**	**O**	
1	1	2		2		1		1.5										**6.5**
2	1	2		2		1		1.5										**6.5**
3	1	2		2		1		1.5										**6.5**
4	2	2		2		1		1.5										**6.5**
5	1	2		2		1		1.5										**6.5**
6	3	2		2		1		1.5			6				1.5			**14**
7	2	2		2		1		1.5			8				2			**16.5**
8	1	2		2		1		1.5										**6.5**
9	3	2		2		1		1.5										**6.5**
10	2	2		2		1		1.5										**6.5**
11	2	2		2		1		1.5										**6.5**
12	3	2		2		1		1.5										**6.5**
13	2	2		2		1		1.5				1.2				0.3		**8**
14	1	2		2		1		1.5										**6.5**
>W14										8				2				**10**
Overall		28		28		14		21		8	14	1.2		2	3.5	0.3		**120**
SLT CREDIT EQUIVALENT																		**3**

9.0 DETAILED SYLLABUS AND TEACHING PLAN

Week	Session	Contents	Reference	Delivery Method

1	**Lecture 1**	**Basic Principles and Definitions** • Requirements analysis and design in the software development lifecycle • Introduction to requirements engineering • Role of communication in requirements engineering • Requirements engineer skills	[1,2,3,]	Lecture
	Lab 1	• Create project group • Create project organization • Define project teams' roles and responsibilities	[1]	Group project/ lab practical
2	**Lecture 2**	**Software Requirements** • Types of requirements • Requirements interaction • The role of requirement quality • Requirements measures	[1,3,4]	Lecture
	Lab 2	• Each group shall find problems to be solved by a software system. • Identify potential software project and name the project or system to be developed. • Write a synopsis of what the system can do. • Identify the system stakeholders	[2]	Group project/ lab practical
3	**Lecture 3**	**Requirements Elicitation** • The purpose and the role of requirements elicitation • Problems with requirements elicitation • Requirements elicitation techniques	[1,3,4]	Lecture

	Lab 3	• Identify one or more elicitation techniques that is suitable for the proposed software project. • Plan and prepare for the elicitation process to elicit the requirements.	[2]	Group project/ lab practical
4	**Lecture 4**	**Requirements Analysis & Negotiation** • Requirements analysis process & techniques • Requirements analysis modeling • Requirements conflicts • Requirements negotiation stages and process	[1,3,4]	Lecture
	Lab 4	• Elicit the requirements according to the technique/s identified. • Follow the analysis process to analyse the elicited requirements. • Exercise requirements negotiation to resolve conflicting requirements.	[2]	Group project/ lab practical
5	**Lecture 5**	**Requirements Documentation using Natural Language** • The importance of requirements specification using natural language. • Language anomaly in natural language. • Technique to write requirements		
	Lab 5	• Document user requirement description of mini project in Software Requirement Specification (SRS) • Develop glossaries based on the mini project. • Capture and use correctly the phrase templates for written statement of user requirement.		

6	**Lecture 6**	**Model-Based Requirements Documentation** • Type of models used for requirements representation. • Models interpretation • Models relationship and interdependencies	[1,3,4]	Lecture Mid Term Test
	Lab 6	• Use case modeling for the mini project. • Interaction diagrams for the mini project.	[2]	Group project/ lab practical
7		MID TERM BREAK		
8	**Lecture 7**	**Requirements Validation** • The purpose and the role of requirements validation • Requirements validation principles • Requirements validation techniques	[1,3,4]	Lecture
	Lab 7	• Validate the requirements in the SRS produced according to the validation techniques.	[2]	Group project/ lab practical
9	**Lecture 8**	**Requirements Management** • Requirements attributes and how to manage them. • Requirements prioritization • Requirement traceability • Requirements change management • Version management • Tool support	[1,3,4]	Lecture
	Lab 8	• Exercise change management to the mini project.	[2]	Group project/ lab practical
10	**Lecture 9**	**An Introduction to Software Design** • Software design overview • Design stages	[1,3,4]	Lecture

		• Design processes • Design principles		
	Lab 9	• Design the class diagram for the mini project	[2]	Group project/ lab practical
11	**Lecture 10**	**High-level Design – Software Architecture** • Overview of Software Architecture • Describe the Software Architecture • How software architecture drives implementation		
	Lab 10	• Design software architecture for the mini project		
12	**Lecture 11**	**Detailed Design Fundamentals** • Overview of detailed design • Key tasks in detailed design	[1,3,4]	Lecture
	Lab 11	• Document the design of the mini project into Software Design Document (SDD)	[2]	Group project/ lab practical
13	**Lecture 12**	**Data Storage and Interface Design** • Types of data and storage • Selection of storage • System interface design • User interface design	[1,3,4]	Lecture
	Lab 12	• Design system data storage and interface for the project	[2]	Group project/ lab practical
14	**Lecture 13**	**Structural and Behavioral Design of Components** • Overview of component design • Designing internal structure of components	[1,3,4]	Lecture

		• Design principles for internal component design		
	Lab 13	• Design the component diagrams for the mini project	[2]	Group project/ lab practical
14	**Lecture 14**	**Industry Talk** • Software requirements and design in practice	[1,3,4]	Lecture
	Lab 14	• Project presentation	[2]	Group project/ lab practical

10.0 MATRIX OF LEARNING OUTCOMES

COURSE vs PROGRAM LEARNING OUTCOME (PLO)

Course	PROGRAM OUTCOME (PO)							
	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8
BITP 2223	X		X				X	

COURSE LEARNING OUTCOME (CLO) vs PROGRAM LEARNING OUTCOME (PLO)

CLO	PROGRAM LEARNING OUTCOME (PLO)							
	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8
CLO1	X							
CLO2							X	
CLO3			X					

COURSE LEARNING OUTCOME (CLO)

CLO1	Analyze software requirement and design the software using object oriented approach (C4)
CLO2	Model software analysis and software design using object oriented approach (P3).
CLO3	Write formal software requirements specification document and software design document (A2, CTPS2)

COURSE vs TAXONOMY

Course	Taxonomy																	
	Affective					Cognitive						Psychomotor						
	A1	A2	A3	A4	A5	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7
BITP 2223	X	X				X	X	X	X			X	X	X				

COURSE LEARNING OUTCOME (LO) vs TAXONOMY

CLO	Taxonomy																	
	Affective					Cognitive						Psychomotor						
	A1	A2	A3	A4	A5	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7
CLO1						X	X	X	X									
CLO2												X	X	X				
CLO3	X	X																

TEACHING PLAN APPROVAL

Prepared by;

……
Name:

Stamp:

Date: _____

Approved by;

……
Dean/Deputy Dean (Academic)/Coordintor/HOD

Stamp:

Date: _____

TEACHING PLAN IMPLEMENTATION (MID SEMESTER BREAK)

Comment :

Checked by;

……..
Dean/Deputy Dean (Academic)/Coordinator/HOD

Stamp :

Date: _____

TEACHING PLAN IMPLEMENTATION (WEEK 16)

Comment :

Checked by;

……..
Dean/Deputy Dean (Academic)/Coordinator/HOD

Stamp :

Date: _____