Template of Course Specification

Faculty: CIT
Department: CS
Program: SWE

I. General information about the course:

1.	Course Title:	Software Engineering				
2.	Course Code and Number :	CS325				
3.	Credit Hours: 3	Lecture	Seminar/Tutorial	Practical	Training	Total
		2		2		3
4.	Study Level and Semester:	Level 3 or 4 / Semester 6 or 7				
5.	Pre-requisites (if any):	- IT & IS: Systems Analysis and Design - SE: Introduction to Database				
6.	Co-requisites (if any):					
7.	Program in which the course is offered	SE, IT, CS, IS				
8.	Teaching Language:	English / Arabic				
9.	Study System :	Course Based				
10.	Prepared by:	Course Facilitator (Fahd N. Alwesabi)				
11.	Approval date :					
12.	Approved by:					

II. Course Description:

This course is very important to computer science, software engineering, IT and IS students in order to takea look at the Software Engineering concepts that are needed to develop software systems that can meet basic functional and non-functional requirements within a well- defined problem domain. It covers the traditional and most recent practices of different software engineering approaches including the requirements analysis, design, development, verification, and maintenance. A look at some related issues such as SWE projects management and critical systems. An examination of object-oriented software construction focused on the Unified Modeling Language.

An introduction to CASE tools. Students will collaboratively define requirements and design of a substantial project. According to the course plan of these programs, this course required the course of system analysis and design as a pre-requisite.

III. CourseAims:

This course aims to:

- Equip student with principles, concepts, methods, and techniques of the software engineering approach to produce quality software (particularly for small, medium, and large systems).
- Equip student with the skills of analysis, develop and implement computer software.
- Enable student to organize and manage a medium-sized software development project, including project plans and documentation, schedule and cost estimates, and quality assurance activities.
- Train student to present effective technical written and oral presentations.
- Enable student to function effectively as a member of a team engaged in technical work by involve him/her in the group-based course project.
- Train student to think critically about ethical and social issues in software engineering.

I. CourseIntended learning outcomes (ILOs):

(A) Knowledge and Understanding:

Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes) in: Knowledge and Understanding Demine.

Knowledge and Understanding PILOs	Knowledge and Understanding CILOs	
After completing this program, students would be able to:	**After completing this course, students would be able to:**	
A1: Understand the principles and theories of mathematics, programming, algorithms and its complexities.	a1	Explain the concepts of software development, testing, and evolution, software process models, techniques, process activities of software requirements engineering (functional and non-functional) and CASE tools
A3: Understand the use and impact of Information and Communication Technology (ICT) by recognizing the concepts of enterprise information management and networking.		
A4. Understand the software process life cycles, for small, medium and large software projects, and different approaches of software development, software project management, risk and quality management, and human-computer interaction	a2	Recognize the principal of software process activities in a general or object-oriented design process, tasks and risks of software project management, software quality management process, quality standards and measurements, different types of software maintenance and the factors that affect maintenance costs.
A5: Recognize the social, professional, legal, ethical, and cultural issues involved in the use of computer technology	a3	Define the key issues of decisions that have to be made about the system architecture and architectural patterns during the architectural design process and key issues that have to be considered when implementing software, including software reuse and open-source development.

(B) Intellectual Skills:

Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes) in: Intellectual skills

Intellectual Skills PILOs	Intellectual Skills CILOs	
After completing this program, students would be able to:	**After completing this course, students would be able to:**	
B1: Determine and analyze criteria, specifications and requirements appropriate to specific computing problems and formulate strategies to solve them.	b1	Systematize the principal requirements engineering activities of elicitation, analysis and validation, and the relationships between these activities.
B2. Differentiate the various methods of computational thinking, determining their wide relevance and applicability within other domains in everyday life, and being able to employ them in different conditions.	b2	Differentiate between agile and plan-driven Development, and propose the appropriate software engineering methods and techniques for the development of each type of software system, Evaluate the legacy systems to decide whether they should be scrapped, maintained, reengineered, or replaced.
B3. Choose the most appropriate methods and tools including deploying appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.		
B4. Criticize systems in terms of general quality attributes and possible tradeoffs presented within the given problem	b3	Organize the non-functional requirements, and define the necessary quality attributes that have to be considered when design and implementing solution for the given problem.

(C) Professional and Practical Skills.

Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes) in: Professional and Practical Skills

Professional and Practical Skills PILOs	Professional and Practical Skills CILOs	
After completing this program, students would be able to:	**After completing this course, students would be able to:**	
C1. Apply fundamental concepts of computer science, software engineering, human-computer interaction, science and mathematics in the modeling and design of computer systems.	c1	Use agile software development methods to develop software system, CASE tools and graphical models to represent, evaluate and manage software project, and use the diagram types in the Unified Modeling

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

C2. Develop a wide range of software through all stages of their life cycle, namely problem analysis, requirements specification, design, construction; verification and evolution using appropriate methods and tools.		Language (UML) for OOA and OOD.
C3. Operate computing equipment and software systems effectively and efficiently		
C4. Evaluate software process and artifacts in terms of software quality and models		
C5. Plan, undertake, and monitor software projects.		

(D) General / Transferable Skills:

Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes) in:

General and Transferable skills

Transferable (General) Skills PILOs	Transferable (General) Skills CILOs	
After completing this program, students would be able to:	**After completing this course, students would be able to:**	
D1. Communication skills: Make succinct presentations to a range of audiences about technical problems and their solutions. This may involve face-to-face, written communication or electronic communication.	d1	Collect and gather software engineering problem requirements, read and write effectively technical reports, present technical ideas, and work in a team of software engineering project taking into account the ethical issues related to software development tasks
D2. Teamwork: Be able to work effectively as a member of a development team.		
D4. Self-management skills: Manage one's own learning and development, including time management and organizational skills.		
D5. Professional development skills: Keep abreast of current developments in the discipline to continue one's own professional development.		

II. Alignment of CILOs to Teaching and Assessment Strategies

First: Alignment of Knowledge and Understanding CILOs

Knowledge and Understanding CILOs		Teaching Strategies	Assessment Strategies
a1	Explain the concepts of software development, testing, and evolution, software process models, techniques, process activities of software requirements engineering (functional and non-functional) and CASE tools	- Lectures and Tutorials - Homework - Classroom and tutorial assignments (problem solving) - Research	- Homework reports - Written exams (Quiz, Midterm and Final Exams). - Interaction with discussions - Research report
a2	Recognize the principal of software process activities in a general or object-oriented design process, tasks and risks of software project management, software quality management process, quality standards and measurements,		

	different types of software maintenance and the factors that affect maintenance costs.		
a3	Define the key issues of decisions that have to be made about the system architecture and architectural patterns during the architectural design process and key issues that have to be considered when implementing software, including software reuse and open-source development.		

Second: Alignment of Intellectual Skills CILOs:			
Intellectual Skills CILOs		**Teaching Strategies**	**Assessment Strategies**
b1	Systematize the principal requirements engineering activities of elicitation, analysis and validation, and the relationships between these activities.	- Lectures and Tutorials - Homework - Classroom and tutorial assignments (problem solving) - Research	- Homework reports - Written exams (Quiz, Midterm and Final Exams). - Research report
b2	Differentiate between agile and plan-driven Development, and propose the appropriate software engineering methods and techniques for the development of each type of software system, Evaluate the legacy systems to decide whether they should be scrapped, maintained, reengineered, or replaced.		
b3	Organize the non-functional requirements, and define the necessary quality attributes that have to be considered when design and implementing solution for the given problem.		

Third: Alignment of Professional and Practical Skills CILOs			
Professional and Practical Skills CILOs		**Teaching Strategies**	**Assessment Strategies**
c1	Use agile software development methods to develop software system, CASE tools and graphical models to represent, evaluate and manage software project, and use the diagram types in the Unified Modeling Language (UML) for OOA and OOD.	- Lectures and Tutorials - Classroom and tutorial assignments (problem solving) - Classroom discussion of real problems (brain storming) - Group-based project.	- Homework reports - Written exams (Quiz, Midterm and Final Exams). - Interaction with discussions - Project report and presentation

Fourth: Alignment of Transferable (General) Skills CILOs			
Transferable (General) Skills CILOs		Teaching Strategies	Assessment Strategies
d1	Collect and gather software engineering problem requirements, read and write effectively technical reports, present technical ideas, and work in a team of software engineering project taking into account the ethical issues related to software development tasks	- Lectures and Tutorials - Homework - Classroom and tutorial assignments (problem solving) - Group-based project.	- Homework reports - Written exams (Quiz, Midterm and Final Exams). - Project report and presentation

III. Course topics and sub-topics (theoretical and practical) with contact hours and alignment to CILOs:

Topics/Units of Course Contents

First: Theoretical Aspects

No.	Course Topics/Units	Sub-topics	No. of Weeks	Contact Hours	CILOs
1	**Introduction**	Definitions, Characteristics of Software, Software Engineering versus other engineering disciplines, Software costs, Software process, CASE, What are the attributes of good software, What are the key challenges facing software engineering?, Professional software development, Software Life Cycle Models, Software engineering ethics, Case studies	1	2	a1, a2, a3, b2, b3, c1, d1
2	**Software Processes**	Software process models, Process activities, Coping with change, and the Rational Unified Process	1	2	a2, a3, b1, b2, b3, c1, d1
3	**Agile Software Development**	Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods	1	2	a1, a2, b1, b2, b3, c1, d1
4	**Requirements Engineering**	Functional and non-functional requirements,	2	4	a2, b1, b2, b3, c1, d1

		The software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management			
5	**System Modeling**	Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering	1	2	a2, a3, b1, b2, b3, c1, d1
6	**Mid Term Exam**	Mid-term exam	1	2	
7	**Architectural Design**	Architectural design decisions, Architectural views, Architectural patterns, Application architectures	1	2	a2, a3, b2, b3, c1, d1
8	**Design and Implementation**	Object-oriented design using the UML, Design patterns, Implementation issues, Open source development	1	2	a2, a3, b2, b3, c1, d1
9	**Software Testing**	Development testing, Test-driven development, Release testing, User testing	1	2	a1, a2, a3, b1, b2, b3, c1
10	**Software Evolution**	Evolution processes, Program evolution dynamics, Software maintenance, Legacy system management	1	2	a1, a2, a3, b1, b2, b3, c1, d1
11	**Project Management and Planning**	Risk management, Managing people, Teamwork, Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques	2	4	a1, a2, b2, c1, d1
12	**Software Quality**	Software quality, Software standards, Reviews and inspections, Software measurement and metrics	1	2	a1, a2, a3, b1, b3, d1
13	**Assignment Presentations**	Assignments presentations.	1	2	ALL
14	**Final Exam**	Final Exam	1	2	ALL
Total number of weeks and hours			**16**	**32**	

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

Second: Practical/Tutorial/Clinical Aspects				
Write up practical/tutorial/clinical topics				
No.	**Practical/Tutorial/Clinical topics**	**No. of Weeks**	**Contact Hours**	**CILOs**
1	CASE tools introduction to feasibility study	**2**	**4**	a1, b2, c1, d1
2	Project Discussion I	**1**	**2**	ALL
3	Agile Software Development: Case Study	**2**	**4**	a1, a2, b1, b2, b3, c1, d1
4	Requirements modeling and Smart Draw	**3**	**6**	a1, a2, b1, b2, b3, c1, d1
5	Architectural design and SmartDraw	**2**	**4**	a1, a2, b1, b2, b3, c1, d1
6	UML analysis and Design	**4**	**8**	a1, a2, b1, b2, b3, c1, d1
7	Project Presentations	**2**	**4**	ALL
Total number of weeks and hours		**16**	**32**	

IV. Teaching strategies:

- Lectures, PPT and lecture notes
- Labs and case studies
- Homework
- Classroom discussion of a real problems (Brainstorming)
- Group-based project.
- Research and self-learning
- Problems Solving
- Classroom Tutorial (Applying case studies using CASE tools, modeling and SmartDraw, UML)

V. Tasks and Assignments:

All topics are integrated and applied during the semester-long group project. The aim of the project is to prepare students for dynamics in a real workplace. Members of the group will meet on a regular basis to discuss the project and to assign individual tasks. Students will be judged primarily on the final project presentations. Milestones of the project that should be submitted are:

- PPT presentation
- Word document as a final report

VI. Learning Assessment:

No.	Assessment Tasks	Week due	Mark	Proportion of Final Assessment	Aligned CILOs
1	Exercises & Homework &	Weekly	10	10%	ALL

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

	Quizzes				
2	Project (group)	Week 15	20	20%	ALL
3	Individual Assignments	Every 3 weeks	5	5%	ALL
4	Interactive class discussion & Research	Weekly	5	5%	ALL
6	Mid-term Exam (theoretical)	Week 8	10	10%	ALL
7	Final Exam (theoretical)	Week 16	50	50%	ALL
	Total		**100**	**100%**	

VII. Learning Resources:

Textbooks-not more than 2

1 Ian Sommerville, 2011, Software Engineering, , Ninth edition, Pearson Education Limited, USA.

Essential References-not less than 4

1 Roger S Pressman, 2007, Software Engineering – A Practitioner's Approach, 7th edition, McGraw Hill, USA.
2 Michael Duck and Richard Read, 2003, Data Communications and Computer Networks, for Computer Scientists and Engineers, 2nd Edition, Pearson Prentice Hall, USA.
3 Timothy C. Lethbridge and Robert Laganière, "Object Practical Software Development using UML and Java", 2nd Edition, McGraw Hill, 2001.
4 Stiller and LeBlanc, 2001, Project-based Software Engineering: an object-oriented approach, , 1st edition, Addison Wesley, USA.
5 Craig Larman, Applying UML and Patterns, Prentice

Electronic Materials and Web Sites

McConnell, Steve. Software Project Survival Guide.
http://www.amazon.com/Software-Project-Survival-Guide-Practices/dp/1572316217
Ludi, Stephanie. *Student Survival Guide to Managing Group Projects* (online):
http://www.csc.calpoly.edu/~sludi/SEmanual/TableOfContents.html

VIII. Course Policies:

1 Class Attendance:

- Attendance in all lectures and practical classes are required, except in very emergency circumstances, such as serious illness or death in the family with providing an acceptable documentation approved by the university and forwarded by the chairman of the department. Otherwise the absence shall be considered unexcused.
- In accordance with the university rules, if the percentage of student's absence exceeds 25 % of the total lectures or practical classes, the student involved shall be disqualified in the final written and practical examination of the course and shall be deemed to have failed in the course.

2 Tardy:

	- Roll will be called in the very beginning of each lecture and practical class. Retardation for more than three weeks without a reasonable cause, the student involved shall not be allowed to attend the class any longer and consequently shall be considered to be absent.
3	**Exam Attendance/Punctuality:** - It is incumbent on student to report at the examination hall for checking in and rolls calling at least 15 minutes before the commencement of examination. - A student is not allowed to submit answer booklet and leave the examination hall only on or after the passage of the half examination duration. - A student who comes late shall not be admitted to the examination hall, only within the first 30 minutes of the examination. After this time, the student will be considered to be missed in the examination and shall be deemed to have failed in the course. - When a student misses the final examination due to a legitimate medical problems or death in the family, an acceptable documentation approved by the university medical unit for the excused absence must be provided no later than three weeks and consequently the student shall be disqualified in the examination but with the excused absence.
4	**Assignments & Projects:** Assignments and reports are to be submitted in hardcopy in due date in the class , unless otherwise that may be specified by the teacher. Email submissions of assignments are not accepted. Late assignments and reports will be penalized at least 5 % per day (including weekends and holidays). Depending on the circumstances the penalty may be higher, for example, if an assignment is received after the solution has been discussed.
5	**Cheating:** If a student is found cheating in examination (midterm or final or quizzes) (copying from unauthorized materials and another students' work or allowing other students to copy from his/her own work), the student involved shall be disqualified in the examination and shall be deemed to have failed in the course and also suspended from examinations of two more courses. If a student is found engaging in any unauthorized communications (oral, sign, call, etc.), while the examination is in progress or in possessing of any authorized materials or electronic devices before the distribution of examination papers , the student involved shall be disqualified in the examination and shall be deemed to have failed the course.
6	**Plagiarism:** Plagiarism is the presentation of any material (text, data or figures) from any other source in preparation of assignments or practical reports without clear and adequate acknowledgement of the source. Plagiarism is also the use or copy of other students' work (with, or without payment) to prepare all or part of undertaken assignments or practical reports of work submitted for assessment. All types of plagiarism are unacceptable and are considered dishonest practices. If a student is found plagiarism, the student involved shall be subjected to the same penalties as in the case of cheating as already mentioned above policies.
7	**Other policies:** Students are expected to be punctual, and, as always, to conduct themselves professionally and courteously: Using electronic devices or speaking with each other is not allowed, the student involved shall be expelled out of the class and shall be considered to be absent

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	.د عبدالقادر العبادي

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	.د عبدالقادر العبادي

Course Syllabus

Faculty: CIT
Department: CS
Program(s): SWE

I. General information about the course instructor :							
Name	Fahd Nasser A. Al-Wesabi	**Office Hours(3 Hours Weekly)**					
Location & phone number	**Sana'a, 60th street**	**Sat**	**Sun**	**Mon**	**Tue**	**Wed**	**Thu**
Email	fwesabi@gmail.com						

IV. General information about the course:						
1.	**Course Title:**	**Software Engineering**				
2.	**Course Code and Number :**	CS325				
3.	**Credit Hours: 3**	**Lecture**	**Seminar/Tutorial**	**Practical**	**Training**	**Total**
		2		2		3
4.	**Study Level and Semester:**	Level 3 or 4 / Semester 6 or 7				
5.	**Pre-requisites (if any):**	- SE: Introduction to Database - IT & IS: Systems Analysis and Design				
6.	**Co-requisites (if any):**					
7.	**Program in which the course is offered**	SE, IT, CS, IS				
8.	**Teaching Language:**	English / Arabic				
9.	**Study System :**	Course Based				
10.	**Prepared by:**	Course Facilitator (Fahd N. Alwesabi)				
11.	**Approval date :**					
12.	**Approved by:**					

I. Course Description:

This course is very important to computer science, software engineering, IT and IS students in order to take a look at the Software Engineering concepts that are needed to develop software systems that can meet basic functional and non-functional requirements within a well- defined problem domain. It covers the traditional and most recent practices of different software engineering approaches including the requirements analysis, design, development, verification, and maintenance. A look at some related issues such as SWE projects management and critical systems. An examination of object-oriented software construction focused on the Unified Modeling Language.
An introduction to CASE tools. Students will collaboratively define requirements and design of

a substantial project. According to the course plan of these programs, this course required the course of system analysis and design as a pre-requisite.

V. Course Aims:

This course aims to:

- Equip student with principles, concepts, methods, and techniques of the software engineering approach to produce quality software (particularly for small, medium, and large systems).
- Equip student with the skills of analysis, develop and implement computer software.
- Enable student to organize and manage a medium-sized software development project, including project plans and documentation, schedule and cost estimates, and quality assurance activities.
- Train student to present effective technical written and oral presentations.
- Enable student to function effectively as a member of a team engaged in technical work by involve him/her in the group-based course project.
- Train student to think critically about ethical and social issues in software engineering.

I. Course Intended Learning Outcomes (CILOs) :

1. Explain the concepts of software development, testing, and evolution, software process models, techniques, process activities of software requirements engineering (functional and non-functional) and CASE tools.
2. Recognize the principal of software process activities in a general or object-oriented design process, tasks and risks of software project management, software quality management process, quality standards and measurements, different types of software maintenance and the factors that affect maintenance costs..
3. Define the key issues of decisions that have to be made about the system architecture and architectural patterns during the architectural design process and key issues that have to be considered when implementing software, including software reuse and open-source development.
4. Systematize the principal requirements engineering activities of elicitation, analysis and validation, and the relationships between these activities.
5. Differentiate between agile and plan-driven Development, and propose the appropriate software engineering methods and techniques for the development of each type of software system, Evaluate the legacy systems to decide whether they should be scrapped, maintained, reengineered, or replaced.
6. Organize the non-functional requirements, and define the necessary quality attributes that have to be considered when design and implementing solution for the given problem.
7. Use agile software development methods to develop software system, CASE tools and graphical models to represent, evaluate and manage software project, and use the diagram types in the Unified Modeling Language (UML) for OOA and OOD.

8. Collect and gather software engineering problem requirements, read and write effectively technical reports, present technical ideas, and work in a team of software engineering project taking into account the ethical issues related to software development tasks

IX. Course topics and sub-topics (theoretical and practical) with contact hours and alignment to CILOs:

Topics/Units of Course Contents

First: Theoretical Aspects

No.	Course Topics/Units	Sub-topics	No. of Weeks	Contact Hours
1	**Introduction**	Definitions, Characteristics of Software, Software Engineering versus other engineering disciplines, Software costs, Software process, CASE, What are the attributes of good software, What are the key challenges facing software engineering?, Professional software development, Software Life Cycle Models, Software engineering ethics, Case studies	1	2
2	**Software Processes**	Software process models, Process activities, Coping with change, and the Rational Unified Process	1	2
3	**Agile Software Development**	Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods	1	2
4	**Requirements Engineering**	Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management	2	4
5	**System Modeling**	Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering	1	2
6	**Mid Term Exam**	Mid-term exam	1	2
7	**Architectural Design**	Architectural design decisions, Architectural views, Architectural patterns, Application architectures	1	2

8	**Design and Implementation**	Object-oriented design using the UML, Design patterns, Implementation issues, Open source development	1	2
9	**Software Testing**	Development testing, Test-driven development, Release testing, User testing	1	2
10	**Software Evolution**	Evolution processes, Program evolution dynamics, Software maintenance, Legacy system management	1	2
11	**Project Management and Planning**	Risk management, Managing people, Teamwork, Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques	2	4
12	**Software Quality**	Software quality, Software standards, Reviews and inspections, Software measurement and metrics	1	2
13	**Assignment Presentations**	Assignments presentations.	1	2
14	**Final Exam**	Final Exam	1	2
Total number of weeks and hours			**16**	**32**

Write up practical/tutorial/clinical topics			
No.	**Practical/Tutorial/Clinical topics**	**No. of Weeks**	**Contact Hours**
1	CASE tools introduction to feasibility study	**2**	**4**
2	Project Discussion I	**1**	**2**
3	Agile Software Development: Case Study	**2**	**4**
4	Requirements modeling and Smart Draw	**3**	**6**
5	Architectural design and SmartDraw	**2**	**4**
6	UML analysis and Design	**4**	**8**
7	Project Presentations	**2**	**4**
Total number of weeks and hours		16	32

X. Teaching strategies:

- Lectures, PPT and lecture notes
- Labs and case studies
- Homework
- Classroom discussion of a real problems (Brainstorming)
- Group-based project.

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

- Research and self-learning
- Problems Solving
- Classroom Tutorial (Applying case studies using CASE tools, modeling and SmartDraw, UML)

XI. Tasks and Assignments:

All topics are integrated and applied during the semester-long group project. The aim of the project is to prepare students for dynamics in a real workplace. Members of the group will meet on a regular basis to discuss the project and to assign individual tasks. Students will be judged primarily on the final project presentations. Milestones of the project that should be submitted are:

- PPT presentation
- Word document as a final report

XII. Learning Assessment:

No.	Assessment Tasks	Week due	Mark	Proportion of Final Assessment
1	Exercises & Homework & Quizzes	Weekly	10	10%
2	Project (group)	Week 15	20	20%
3	Individual Assignments	Every 3 weeks	5	5%
4	Interactive class discussion & Research	Weekly	5	5%
6	Mid-term Exam (theoretical)	Week 8	10	10%
7	Final Exam (theoretical)	Week 16	50	50%
	Total		**100**	**100%**

XIII. Learning Resources:

Textbooks-not more than 2

2 Ian Sommerville, 2011, Software Engineering, , Ninth edition, Pearson Education Limited, USA.

Essential References-not less than 4

6 Roger S Pressman, 2007, Software Engineering – A Practitioner's Approach, 7th edition, McGraw Hill, USA.

7 Michael Duck and Richard Read, 2003, Data Communications and Computer Networks, for Computer Scientists and Engineers, 2nd Edition, Pearson Prentice Hall, USA.

8 Timothy C. Lethbridge and Robert Laganière, "Object Practical Software Development using UML and Java", 2nd Edition, McGraw Hill, 2001.

9 Stiller and LeBlanc, 2001, Project-based Software Engineering: an object-oriented approach, , 1st edition, Addison Wesley, USA.

10 Craig Larman, Applying UML and Patterns, Prentice

الموصف	المراجع	مسؤول البرنامج	رئيس القسم	عميد الكلية
		أ.وديع القباطي	أ.نبيل المخلافي	د. عبدالقادر العبادي

Electronic Materials and Web Sites	
	McConnell, Steve. Software Project Survival Guide. http://www.amazon.com/Software-Project-Survival-Guide-Practices/dp/1572316217 Ludi, Stephanie. *Student Survival Guide to Managing Group Projects* (online): http://www.csc.calpoly.edu/~sludi/SEmanual/TableOfContents.html

XIV. Course Policies:

1	**Class Attendance:** - Attendance in all lectures and practical classes are required, except in very emergency circumstances, such as serious illness or death in the family with providing an acceptable documentation approved by the university and forwarded by the chairman of the department. Otherwise the absence shall be considered unexcused. - In accordance with the university rules, if the percentage of student's absence exceeds 25 % of the total lectures or practical classes, the student involved shall be disqualified in the final written and practical examination of the course and shall be deemed to have failed in the course.
2	**Tardy:** - Roll will be called in the very beginning of each lecture and practical class. Retardation for more than three weeks without a reasonable cause, the student involved shall not be allowed to attend the class any longer and consequently shall be considered to be absent.
3	**Exam Attendance/Punctuality:** - It is incumbent on student to report at the examination hall for checking in and rolls calling at least 15 minutes before the commencement of examination. - A student is not allowed to submit answer booklet and leave the examination hall only on or after the passage of the half examination duration. - A student who comes late shall not be admitted to the examination hall, only within the first 30 minutes of the examination. After this time, the student will be considered to be missed in the examination and shall be deemed to have failed in the course. - When a student misses the final examination due to a legitimate medical problems or death in the family, an acceptable documentation approved by the university medical unit for the excused absence must be provided no later than three weeks and consequently the student shall be disqualified in the examination but with the excused absence.
4	**Assignments & Projects:** Assignments and reports are to be submitted in hardcopy in due date in the class , unless otherwise that may be specified by the teacher. Email submissions of assignments are not accepted. Late assignments and reports will be penalized at least 5 % per day (including weekends and holidays). Depending on the circumstances the penalty may be higher, for example, if an assignment is received after the solution has been discussed.
5	**Cheating:** If a student is found cheating in examination (midterm or final or quizzes) (copying from unauthorized materials and another students' work or allowing other students to copy from his/her own work), the student involved shall be disqualified in the examination and

	shall be deemed to have failed in the course and also suspended from examinations of two more courses. If a student is found engaging in any unauthorized communications (oral, sign, call, etc.), while the examination is in progress or in possessing of any authorized materials or electronic devices before the distribution of examination papers , the student involved shall be disqualified in the examination and shall be deemed to have failed the course.
6	**Plagiarism:** Plagiarism is the presentation of any material (text, data or figures) from any other source in preparation of assignments or practical reports without clear and adequate acknowledgement of the source. Plagiarism is also the use or copy of other students' work (with, or without payment) to prepare all or part of undertaken assignments or practical reports of work submitted for assessment. All types of plagiarism are unacceptable and are considered dishonest practices. If a student is found plagiarism, the student involved shall be subjected to the same penalties as in the case of cheating as already mentioned above policies.
7	**Other policies**: Students are expected to be punctual, and, as always, to conduct themselves professionally and courteously: Using electronic devices or speaking with each other is not allowed, the student involved shall be expelled out of the class and shall be considered to be absent