



University: *Fayoum University*
Faculty: *Computers and Information*
Department: *Computer Science*
General Diploma

Course Specification

1- Basic Information							
Code: INF 503	Course Title: Software Engineering						
Programme: General Diploma in Computer Science	Number of units: <table style="display: inline-table; vertical-align: middle; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">Lecture:</td> <td style="border: 1px solid black; padding: 2px 5px;">2 hrs/ week</td> </tr> <tr> <td style="padding: 2px 5px;">Tutorial:</td> <td style="border: 1px solid black; padding: 2px 5px;">0 hrs/ week</td> </tr> <tr> <td style="padding: 2px 5px;">Practical:</td> <td style="border: 1px solid black; padding: 2px 5px;">2 hrs/ week</td> </tr> </table>	Lecture:	2 hrs/ week	Tutorial:	0 hrs/ week	Practical:	2 hrs/ week
Lecture:	2 hrs/ week						
Tutorial:	0 hrs/ week						
Practical:	2 hrs/ week						

2- Aims of Course:	<ol style="list-style-type: none"> 1. To understand Software Engineering. 2. To gain basic knowledge software process and software life cycle. 3. To be able to achieve different software artifacts in different software development phases.
---------------------------	---

3- Intended Learning Outcomes	
A- Knowledge and Understanding:	<ol style="list-style-type: none"> a1) Understanding software process and software evolution. a2) Recognize the different software models.
B- Intellectual Skills:	<ol style="list-style-type: none"> b1) Understand difference between Software Engineering and Computer Engineering. b2) Understand costs of Software Engineering. b3) Evaluating attributes of good software.
C- Professional and Practical Skills:	<ol style="list-style-type: none"> c1) Practicing software artifacts/documents development. c2) Practicing system design and system decomposition. c3) Practicing requirements engineering. c4) Working with CASE/UML tools.
D- General and transferable Skills	<ol style="list-style-type: none"> d1) Practicing working in teams d2) Attending Competitions with colleagues

4-Course Content:	<ol style="list-style-type: none"> 1. Introduction: Well-engineered software, the software process, software evolution, and software reliability. 2. Human factors in software engineering: Human diversity, knowledge processing, group working. 3. Software specification and system modeling: The software requirements document 4. Traditional Processes 5. Requirements definition and specification: Requirements specification, nonfunctional requirements definition. 6. Requirements validation and prototyping: The prototyping process, prototyping techniques. 7. Software design: Top-down design, systems design, design decomposition, 8. software design quality 9. Software Testing Methods 10. Software Reuse 11. Agile Methods
--------------------------	--

5- Teaching and Learning Methods:	<ol style="list-style-type: none"> 1. Lectures 2. Computer-lab Sessions 3. Practical lab work 4. Class discussions 5. Internet searches 6. Group projects
--	---

6- Teaching and Learning Methods for handicapped students :	-
--	---

7- Student Assessment	
A- Assessment Methods:	<ol style="list-style-type: none"> 1. Assignments and Quizzes 2. Midterm written exam 3. Oral exam 4. Practical exam 5. Final written exam
B- Assessment schedule:	Midterm Examination: Week 6 Practical examination: Week 12 Oral Examination: Week 12 Final Examination: Week 15
C- Weighting of assessments:	Assignments: 15% Mid-Term Examination: 15% Oral Examination: 0% Practical Examination: 10% Final-term Examination: 60%

8- Books and References	
A- Notes:	Carefully selected, well reviewed in core presentations
B- Essential Books (Text Books):	<ul style="list-style-type: none"> ▪ Ian Sommerville, "Software Engineering", Addison-Wesley publisher.

<p>C- Recommended Books:</p>	<ul style="list-style-type: none"> ▪ Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: “Design Patterns”, Addison-Wesley. ▪ Grady Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language User Guide”, Addison Wesley. ▪ Bernd Bruegge, Allen Dutiot: “Object oriented Software engineering using UML, Patterns and Java”, Prentice Hall.
<p>D- Periodicals, Web sites, ... etc</p>	<p>-</p>

Course Professor: **Department Head:**

Course Content Intended Learning Outcomes Matrix

Course Title: Software Engineering

Course Code INF 503

Course Content	Week	Knowledge & Understanding		Intellectual Skills			Professional & Practical Skills				General & Transferable Skills	
		a1	a2	b1	b2	b3	c1	c2	c3	c4	d1	d2
1. Introduction: Well-engineered software, the software process, software evolution, and software reliability.	1	x		x		x	x				x	x
2. Human factors in software engineering: Human diversity, knowledge processing, group working.	2	x				x					x	x
3. Software specification and system modeling: The software requirements document, requirements evolution, system contexts,	3	x	x				x		x		x	x
4. Traditional Processes.	4	x	x						x		x	x
5. Requirements definition and specification: Requirements specification, nonfunctional requirements definition.	5	x			x				x		x	x
6. Requirements validation and prototyping: The prototyping process, prototyping techniques.	6	x							x		x	x
7. Software design: Top-down design, systems design, design decomposition,	8	x	x					x			x	x
8. software design quality,	9				x	x					x	x
9. Software Testing Methods	10	x	x				x	x		x	x	x
10. Software Reuse	11	x	x				x				x	x
11. Agile Methods	12	x	x				x				x	x

Course coordinator:

Head of Department: