



Arab Academy for Science and Technology & Maritime Transport

College of Computing and Information Technology, Cairo

Department of Software Engineering

# **Bachelor of Science in Software Engineering**

## **Program Specification**

**(Year 2013- 2014)**

### **A. Basic Information**

1. Program title: Bachelor of Science in Software Engineering
2. Program type: Single
3. Department responsible for the program: Software Engineering
4. Date of program approval: 2009

## B. Professional Information

Since the dawn of computing in the 1940s, the applications and uses of computers have grown at a staggering rate. Software plays a central role in almost all aspects of daily life. The number, size, and application domains of computer programs have grown dramatically; as a result, hundreds of billions are being spent on software development, and the livelihood and lives of most people depend on the effectiveness of this development. Despite these successes, there are serious problems in the cost, timeliness, and quality of many software products. The reasons for these problems are many and include the following:

- Software products are among the most complex of man-made systems, and software by its very nature has intrinsic, essential properties (e.g., complexity, invisibility, and changeability) that are not easily addressed.
- Programming techniques and processes that worked effectively for an individual or a small team to develop modest-sized programs do not scale-up well to the development of large, complex systems (i.e., systems with millions of lines of code, requiring years of work, by hundreds of software developers).
- The pace of change in computer and software technology drives the demand for new and evolved software products. This situation has created customer expectations and competitive forces that strain our ability to produce quality of software within acceptable development schedules.

The field of Software Engineering addresses these issues.

### 1. Program General Aims

The program aims to provide the student with both breadth and depth of knowledge in the concepts and techniques related to the design, programming, and application of large-scale complex computing systems.

After successfully completing the Software Engineering program, the graduate should be able to:

- O1. Development of the ability to recognize problems that are amenable to software engineering, and knowledge of the tools necessary for solving such problems.
- O2. Understand fundamentals of software development life cycle (SDLC), design and analysis, information security, data mining, and crisis management.
- O3. Understand and apply a wide range of principles and tools of software engineering, such as design methodologies, choice of algorithm, language, software libraries and user interface technique.
- O4. Implement solutions, including use of appropriate programming languages, web-based systems and tools, design methodologies, and database systems.
- O5. Design, implement, and evaluate a computer-based systems, process, component or program.

- O6. Use knowledge and understanding in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.
- O7. Specify, design, and implement computer-based information systems, and evaluate them in terms of general quality attributes and possible tradeoffs presented within the given problem.
- O8. Perform troubleshooting in software systems.
- O9. Describe characteristics of various components of information systems, use the appropriate tools and techniques to analyze, design, and construct information systems.
- O10. Communicate effectively by oral, written and visual means.
- O11. Work effectively as an individual and as a member of a team in designing and implementing software systems.
- O12. Perform independent and efficient time management.
- O13. Analyze the local and global impact of computing on individuals, organizations, and society.
- O14. Aware of key ethical issues affecting information systems and their responsibilities as information science professionals.
- O15. Demonstrate the ability to effectively communicate ideas and concepts by oral, written, and visual means clearly and in an organized manner.

## 1. Intended Learning outcomes (ILOs)

### a. Knowledge and Understanding

**The graduates of the software engineering program should be able to demonstrate knowledge and understanding of:**

- K1. Essential facts, concepts, principles and theories relating to computing and information and computer applications as appropriate to the program of study.
- K2. Modeling and design of computer-based systems bearing in mind the trade-offs.
- K3. Tools, practices and methodologies used in the specification, design, implementation and evaluation of computer software systems.
- K4. Criteria and specifications appropriate to specific problems, and plan strategies for their solution.
- K5. The extent to which a computer-based system meets the criteria defined for its current use and future development.
- K6. The current and underlying technologies that support computer processing and inter-computer communication.
- K7. Principals of generating tests which investigate the functionality of computer programs and computer systems and evaluating their results.

- K8. Management and economics principles relevant to computing and information disciplines.
- K9. Professional, moral and ethical issues involved in the exploitation of computer technology and be guided by the appropriate professional, ethical and legal practices relevant to the computing and information industry.
- K10. Current developments in computing and information research.
- K11. Requirements, practical constraints and computer-based systems.
- K12. Understanding essential facts, concepts, principles and theories relevant to software engineering.
- K13. Demonstrate basic knowledge and understanding of a core of analysis, algebra, applied mathematics and statistics.
- K14. Use high-level programming languages.
- K15. Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models.
- K16. Know and understand the principles and techniques of database management systems, management, data mining, multimedia, application development, business process management, human-computer interaction, object-oriented analysis and design, e-technologies, multimedia, software security.
- K17. Show a critical understanding of the broad context within software engineering including issues such as quality, reliability.
- K18. Understand the principles of Information communication and information security.
- K19. Perform specification, analysis, design, implementation and testing of software solutions.
- K20. Modeling organizational processes and data, defining and implementing technical and process solutions, managing projects, and integrating software systems
- K21. Types and alternatives of software systems architectures, and their differences in terms of performance, cost consequences, and their implications for the software quality attributes needed.
- K22. Understand the challenges inherent in the maintenance and evolution of software systems, and the techniques and best practices currently available for dealing with them.

## **b. Intellectual Skills**

**The graduates of the software engineering program should be able to:**

- I1. Analyze computing problems and provide solutions related to the design and construction of computing systems.

- I2. Realize the concepts, principles, theories and practices behind computing and information as an academic discipline.
- I3. Identify criteria to measure and interpret the appropriateness of a computer system for its current deployment and future evolution.
- I4. Analyze, propose and evaluate alternative computer systems and processes taking into account limitations, and quality constraints.
- I5. Make ideas, proposals and designs using rational and reasoned arguments for presentation of computing systems.
- I6. Evaluate the results of tests to investigate the functionality of computer systems.
- I7. Achieve judgments considering balanced costs, benefits, safety, quality, reliability, and environmental impact.
- I8. Be familiar with the professional, legal, moral and ethical issues relevant to the computing industry.
- I9. Evaluate research papers in a range of knowledge areas
- I10. Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results.
- I11. Perform comparisons between (methods, techniques, strategies ...etc).
- I12. Identify attributes, components, relationships, patterns, main ideas, and errors.
- I13. Restrict solution methodologies upon their results.
- I14. Select the suitable tools, methods and techniques for modeling, analyzing software, establishing criteria, and verify solutions.
- I15. Identify a range of solutions and critically evaluate and justify proposed design solutions.
- I16. Solve software problems with pressing commercial, time, and industrial constraints.
- I17. Generate an innovative design to solve a problem containing a range of commercial and industrial constraints.
- I18. Perform problem analysis from written descriptions; derive requirements specifications from an understanding of problems (analysis, synthesis).
- I19. Create and/or justify designs to satisfy given requirements (synthesis, evaluation, application);

### **c. Practical and Professional Skills**

**The graduates of the software engineering program should be able to:**

- P1. Operate computing equipment, recognizing its logical and physical properties, capabilities and limitations.
- P2. Implement comprehensive computing knowledge and skills in projects and in deployment of computers to solve position practical problems.
- P3. Deploy the equipment and tools used for the construction, maintenance and

documentation of computer applications.

- P4. Apply computing information retrieval skills in computing community environment and industry.
- P5. Develop a range of fundamental research skills, through the use of online resources, technical repositories and library-based material
- P6. Design, implement, maintain, and manage software systems.
- P7. Assess the implications, risks or safety aspects involved in the operation of computing equipment within a specific context.
- P8. Handle a mass of diverse data, assess risk and draw conclusions.
- P9. Use appropriate programming languages, web-based systems and tools, design methodologies, and database systems.
- P10. Use quantitative analysis techniques appropriately and effectively
- P11. Plan and manage a software system project from inception to final implementation and cut-over.
- P12. Perform independent information acquisition and management, using the scientific literature and Web sources.
- P13. Communicate effectively by oral, written and visual means, produce acceptable reports and technical and user system documentation.
- P14. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.
- P15. Using tools to automate software development phases.
- P16. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria
- P17. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.
- P18. Maintaining existing software systems
- P19. Identify any risks or safety aspects that may be involved in the operation of computing equipment within a given context.
- P20. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.
- P21. Prepare technical reports, and a dissertation, to a professional standard.

#### **d. General and Transferable Skills**

**The graduates of the software engineering program should be able to:**

- G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning.
- G2. Demonstrate skills in group working, team management, time management and organizational skills.

- G3. Show the use of information-retrieval.
- G4. Use an appropriate mix of tools and aids in preparing and presenting reports for a range of audiences, including management, technical, users, industry or the academic community.
- G5. Exhibit appropriate numeracy skills in understanding and presenting cases involving a quantitative dimension.
- G6. Reveal communication skills, public speaking and presentation skills, and delegation, writing skills, oral delivery, and effectively using various media for a variety of audiences.
- G7. Show the use of general computing facilities.
- G8. Demonstrate an appreciation of the need to continue professional development in recognition of the requirement for life-long learning.

### **3. Program Academic Standards**

- 1- National Academic Reference Standards (NARS) for Computing and Information, October 2010.
- 2- Standards of Higher Supreme Education Council for the Egyptian Universities.
- 3- Standards of ACM/IEEE CS curricula 2013 (Ironman report) .

### **4. Reference indices (Benchmarks)**

1. Handbook for Academic Review (QAA 2000): Users Guide to the Academic Review of Subjects in Higher Education Institutions in the Transitional Period, 2002-2005.
2. Handbook of Institutional Audit, England, 2002.
3. Collaborative Provision Audit: Supplement to the Handbook for Institutional Audit : England, December, 2004.
4. Nick Harris, Quality Assurance and Accreditation, Report of Visit, March 2008, U.K.

## 5. Curriculum structure and contents of program

**5.A Program duration:** 8 semesters.

### 5.B Program structure:

Total Credits : 144 credit hours

Mandatory Credits: 117 credit hours

Elective Credits: 27 Credit hours

**1- General Requirements (Humanities):**

- 15 credit hours. (10%)  
12 mandatory credit hours.  
3 elective credit hours.

**2- College Requirements:**

- 75 mandatory credit hours. (53%)

**3- Major specialization Requirements: (29%)**

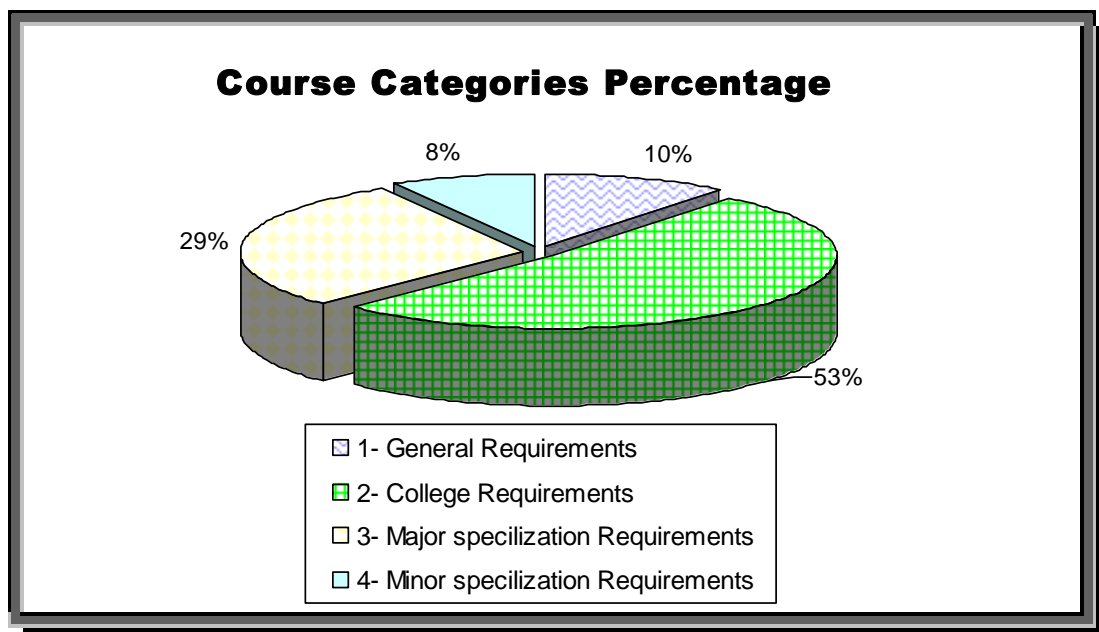
- 30 mandatory credit hours.
- 12 elective credit hours from the list of major elective courses.

**4- Minor specialization Requirements: (8%)**

- 12 elective credit hours from the lists of minor elective courses.

**5- Professional Training:**

- 4 courses in one of the available tracks and their courses are not counted towards the credit requirements.





**5.C Program levels (in credit-hours system):**

Level 1/Year 1: Required to pass 12 courses (distributed as follows):

Compulsory 12 courses      Elective 0      Optional 0

Level 1/Year 2: Required to pass 12 courses (distributed as follows):

Compulsory 12 courses      Elective 0      Optional 0

Level 1/Year 3: Required to pass 12 courses (distributed as follows):

Compulsory 8 courses      Elective 4 courses      Optional 0

Level 1/Year 4: Required to pass 12 courses (distributed as follows):

Compulsory 8 courses      Elective 4 courses      Optional 0

**Program Detailed Structure**

Term I			
Course		Prerequisite	
Code	Title	Code	Title
LH135	ESP-I	LH130**	ESP 0
BA101	Calculus I	BA003 *	Math 0
BA113	Physics		
AR115	Visual Studies		
CS111	Introduction to Computers		
IS171	Introduction to Information systems		
BA003 *	Math 0		
LH130**	ESP 0		
Term II			
Course		Prerequisite	
Code	Title	Code	Title
LH136	ESP-II	LH135	ESP-I
BA102	Calculus II	BA101	Calculus I
NC272	Fundamentals of Business		
EC134	Fundamentals of Electricity and Electronics	BA113	Physics
CS143	Introduction to Problem Solving and Programming	CS111	Introduction to Computers
NC233	Communication Skills		

(\*): This course is added to term I students of science section.

(\*\*) This course is for students who failed the English assessment test

Term III			
Course		Prerequisite	
Code	Title	Code	Title
CE216	Digital Logic Design	EC134	Fundamentals of Electricity and Electronics
CS243	Object-Oriented Programming	CS143	Introduction to Problem Solving and Programming
BA201	Calculus III	BA102	Calculus II
CS202	Discrete Structures	CS111	Introduction to Computers
BA203	Probability and Statistics	BA102	Calculus II
	Humanities Elective		
Term IV			
Course		Course	
Code	Code	Code	Code
SE291	Introduction to Software Engineering	CS243	Object-Oriented Programming
		IS171	Intro. to Information systems
CS212	Data Structures and Algorithms	CS243	Object-Oriented Programming
IS373	Database systems	CS143	Intro. to Problem Solving and Programming
BA204	Linear Algebra	BA102	Calculus II
CE243	Intro. to Computer Architecture	CE216	Digital Logic Design
CE231	Introduction to Networks	CS143	Intro. to Problem Solving and Programming
Term V			
Course		Prerequisite	
Code	Title	Code	Title
SE392	Software requirement & Specifications	SE291	Intro. to Software Engineering
CS244	Advanced Programming Applications	CS243	Object-Oriented Programming
SE391	Project Management	SE291	Intro. to Software Engineering
CS311	Theory of Computation	CS202	Discrete Structures
CS433	Web Programming	IS373	Database Systems
	Minor Elective		

<b>Term VI</b>			
<b>Course</b>		<b>Prerequisite</b>	
<b>Code</b>	<b>Title</b>	<b>Code</b>	<b>Title</b>
<b>CS322</b>	Operating Systems	<b>CE243</b>	Intro. to Computer Architecture
<b>CS366</b>	Introduction to Artificial Intelligence	<b>CS212</b> <b>CS202</b>	Data Structures & Algorithms Discrete Structures
<b>CS312</b>	Computing Algorithms	<b>CS212</b> <b>CS311</b>	Data Structures & Algorithms Theory of Computations
<b>SE393</b>	Principles of Software Architecture	<b>SE291</b>	Intro. to Software Engineering
<b>CS451</b>	Human Computer Interaction	<b>SE291</b>	Intro. to Software Engineering
	Minor Elective		
<b>Term VII</b>			
<b>Course</b>		<b>Prerequisite</b>	
<b>Code</b>	<b>Title</b>	<b>Code</b>	<b>Title</b>
<b>SE491</b>	Software component Design	<b>SE393</b>	Principle of software Architecture
<b>CS481</b>	Computers & Society		99 CR or more
<b>SE401</b>	Project I		GPA=2.0 & 99 CR or more
	Major Elective		
	Major Elective		
	Minor Elective		
<b>Term VIII</b>			
<b>Course</b>		<b>Prerequisite</b>	
<b>Code</b>	<b>Title</b>	<b>Code</b>	<b>Title</b>
<b>SE492</b>	Software verification	<b>SE291</b>	Introduction to Software Engineering
<b>CS421</b>	Computer System Security	<b>CS322</b> <b>CE231</b>	Operating Systems Introduction to Networks
<b>SE402</b>	Project II		GPA=2.0 & 117 CR or more
	Major Elective		
	Major Elective		
	Minor Elective		

**Courses for Major Electives**

Code	Title	Prerequisite	
		Code	Title
SE493	Software Quality Assurance	SE291	Introduction to Software Engineering
SE494	Formal Methods in Software Engineering	SE291	Introduction to Software Engineering
SE495	Security in Software Engineering	SE291	Introduction to Software Engineering
CS443	Game Programming	CS243 CS452	Object-oriented Programming Computer Graphics
CS427	Embedded Systems Programming	CE243 CS143	Introduction to Computer Architecture Introduction to Problem Solving
SE496	Software Engineering Process	SE291	Introduction to Software Engineering

**Courses for Minor Electives**

_Code	Title	Prerequisite	
		Code	Title
CS301	Numerical Methods	BA204 CS142	Linear Algebra Introduction to Problem Solving & Programming
CS345	Structure of Programming Languages	CS311 CS321	Theory of Computation, System Programming
CS405	System Modeling & Simulation	BA203	Probability and Statistics, 99 Hours
CS452	Computer Graphics	CS212 BA204	Data Structures and Algorithms, Linear Algebra
CS321	Systems Programming	CS243 CE243	Object-Oriented Programming Introduction to Computer Architecture
IS372	Information Systems Theory & Practice	IS171	Introduction to Information Systems
IS391	System Analysis & Design	IS171 CS243	Introduction to Information systems Object-Oriented Programming
IS461	Decision Support Systems	CS366	Introduction to Artificial Intelligence
IS474	Advanced Database Systems	IS373	Database Systems

### **Elective Humanities Courses**

<b>Code</b>	<b>Title</b>
<b>NC252</b>	Principles of Marketing
<b>NC264</b>	Principle of Microeconomics
<b>NC273</b>	Global Business
<b>NC282</b>	Financial Accounting

## 5. D Program courses

### 5.D.1. Mandatory General Requirements (12 Credit Hours = 8.33%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
LH135	ENGLISH FOR SPECIAL PURPOSES I	2	2	0	3	X								
LH136	ENGLISH FOR SPECIAL PURPOSES II	2	2	0	3		X							
NC233	Communication Skills	2	2	0	3		X							
NC272	Fundamentals of Business	2	2	0	3		X							

### 5.D.2. Elective General Requirements (3 Credit Hours = 2.01%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
NC264	Principle of Microeconomics	2	2	0	3			X						
NC273	Global Business	2	2	0	3			X						
NC282	Financial Accounting	2	2	0	3			X						
NC252	Principles of Marketing	2	2	0	3			X						

### 5.D.3. College Requirements (75 Credit Hours = 52.08%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
BA101	Calculus I	2	2	0	3	X								
CS111	Introduction to Computers	2	2	2	3	X								
IS171	Introduction to Information Systems	2	0	2	3	X								
BA113	Physics	2	2	2	3	X								

AR115	Visual Studies	2	2	0	3	X												
BA102	Calculus II	2	2	0	3		X											
EC134	Fundamentals of Electricity & Electronics	2	2	2	3		X											
CS143	Intro. to Problem Solving and Programming	2	2	2	3		X											
CE216	Digital Logic Design	2	2	2	3			X										
CS243	Object-Oriented Programming	2	2	2	3			X										
BA201	Calculus III	2	2	0	3			X										
CS202	Discrete Structures	2	2	0	3			X										
BA203	Probability and Statistics	2	2	0	3			X										
BA204	Linear Algebra	2	2	0	3				X									
CS212	Data Structures and Algorithms	2	0	2	3				X									
CE231	Introduction to Networks	2	2	2	3				X									
CE243	Introduction to Computer Architecture	2	2	0	3				X									
IS373	Database Systems	2	0	2	3				X									
CS244	Advanced Programming Applications	2	2	2	3					X								
CS322	Operating Systems	2	2	2	3						X							
CS366	Introduction to Artificial Intelligence	2	2	0	3							X						
CS433	Web Programming	2	0	2	3												X	
SE401	Project I	2	2	0	3													X
CS481	Computers and Society	2	0	2	3													X
SE402	Project II	2	2	0	3													X

## 5.D.4. Major SE Requirements

## 5.D.4.1 Mandatory Courses (30 Credit Hours = 20.83%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
SE291	Introduction to Software Engineering	2	2	0	3				X					
SE392	Software Requirements & Specifications	2	2	0	3					X				
SE391	Project Management	2	0	2	3					X				
CS311	Theory of Computation	2	2	0	3					X				
CS312	Computing Algorithms	2	2	0	3						X			
CS451	Human Computer Interaction	2	0	2	3							X		
SE393	Principles of Software Architecture	2	2	0	3						X			
SE491	Software Component Design	2	0	2	3							X		
CS421	Computer System Security	2	2	0	3								X	
SE492	Software Verification	2	0	2	3									X

## 5.D.4.2 Major Elective Courses (12 Credit Hours = 8.3%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
SE493	Software Quality Assurance	2	2	0	3									X
SE494	Formal Methods in Software Engineering	2	2	0	3								X	
SE495	Security in Software Engineering	2	2	0	3								X	
CS443	Game Programming	2	0	2	3									X
CS427	Embedded Systems Programming	2	0	2	3					X				
SE496	Software Engineering Process	2	2	0	3						X			



## 5.D.5. Minor Requirements (12 Credit Hours = 8.3%)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
IS372	Information Systems Theory & Practice	2	0	2	3					X				
IS474	Advanced Database Systems	2	0	2	3						X			
IS391	System Analysis & Design	2	0	2	3							X		
IS461	Decision Support Systems	2	0	2	3								X	
CS301	Numerical Methods	2	2	0	3					X				
CS345	Structure of Programming Languages	2	2	0	3						X			
CS405	System Modeling & Simulation	2	2	0	3								X	
CS452	Computer Graphics	2	0	2	3							X		
CS321	Systems Programming	2	2	0	3							X		

## 5.D.6. Professional Training (0 Credit Hours)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
	Oracle: OCA (Oracle Certified Forms Developer)	0	2	2	0					X	X	X		
	Cisco: CCNA	0	2	2	0					X	X	X		

## 5.D.7. Practical Training (0 Credit Hours)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
IT400	Summer Training	0	0	0	0							X	X	

## 5.D.8. Preparatory English Course (0 Credit Hours)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
LH130	ENGLISH FOR SPECIAL PURPOSES 0	2	2	0	0	X								

## 5.D.9. Preparatory Math Course (0 Credit Hours)

Code	Title	Contact Hours			Hours	Semester								
		Lecture	Tutorial	Lab		1	2	3	4	5	6	7	8	
BA003	Math 0	4	4	0	0	X								

## 5.D.10. Indicative curricula content by subject area

	Subject Area	Percentage of time spent	Tolerance
<b>A</b>	Humanities, ethical and Social Sciences LH135, LH136, NC233, NC272, 1 elective	9.3 %	8-10
<b>B</b>	Mathematics and Basic Sciences BA101, BA113, BA102, EC134, BA201, BA203, BA204, AR115	14.8%	16-18
<b>C</b>	Basic Computing Sciences CS111, IS171, CS143, CE216, CS243, CS202, CS212, CE231, CE311, IS373, SE291, CS244, CS322, CS366, CS433, CS481	29.6%	26-28
<b>D</b>	Applied Software Engineering (Specialization) SE392, SE391, CS311, CS312, CS451, SE393, SE491, CS421, SE492 and 8 electives	31.4%	28-30
<b>E</b>	Training Summer training (1 month = about 175 hrs) == 2 courses	3.7 %	3-5
<b>F</b>	Projects SE401, SE402	3.7%	3-5
	Subtotal	92.5 %	84-96
<b>G</b>	Optional 4 Prof Training COURSES	7.4 %	4-16
	Total	100%	100

## 6. Contents of Courses

**Course Code:** LH135

**Course Title:** ENGLISH FOR SPECIAL PURPOSES-I

**Course Contents:**

- Gain a sufficient stock of computing and IT vocabulary.
- Read technical texts effectively and apply essential reading skills.
- Employ different language functions and grammatical structures.
- Use technical terms in producing varied writing tasks.
- Listen to interviews, discussions and lectures on computing and IT topics for gist and specific information and also take down notes.
- Write coherent and well-developed paragraphs.
- Engage in discussions and different speaking activities.

**Course Code:** LH136

**Course Title:** ENGLISH FOR SPECIAL PURPOSES-II

**Course Contents:**

- Increase their stock of essential computing and IT vocabulary.
- Read authentic technical texts effectively and apply essential reading skills.
- Employ different Language functions and grammatical structures.
- Use technical terms to produce varied tasks.
- Listen to interviews, discussion and lectures on computing and IT topics for gist and specific information and also take down notes.
- Write coherent and well-developed essays.
- Engage in discussions and different speaking activities.

**Course Code:** NC233

**Course Title:** Communication Skills

**Course Contents:**

- Package their skills in a job-getting interview and team work communication
- Identify and focus on the main message and content
- Organize speaking and writing in a logical, professional-looking fashion
- Evaluate information resources
- Avoid plagiarism by adhering to correct referencing principles
- Think critically
- Speak confidently and authoritatively, with or without, speech notes
- Communicate through public presentations

**Course Code: BA101**

**Course Title: Calculus I**

**Course Contents:**

- Differentiate certain types of functions (trigonometric functions and their inverse, exponential function, and logarithmic function).
- Understand and use the applications of differentiation (l'Hopital, Taylor and Maclaurin's expansions).

**Course Code: BA102**

**Course Title: Calculus II**

**Course Contents:**

- Integrate different types of functions using various methods (Integrating by substitution, integration by parts).
- Understand and use the applications of integration (Area & volume, average of a function, length of a curve).
- Solve system of linear equations using Matrices, and determinants.

**Course Code: BA201**

**Course Title: Calculus III**

**Course Contents:**

- Use the Laplace transform and the theorems (first shift theorem, transform of differentiation and integration theorems, etc....) in solving differential and integral equations.
- Understand the Fourier analysis which includes the Fourier series and Fourier transform.
- Know the concept of linear programming in order to solve system of linear inequalities using the simplex method.

**Course Code: BA203**

**Course Title: Probability and Statistics**

**Course Contents:**

- Make statistical analysis and calculating statistical measurements using computer programs like the Minitab program or Excel.
- Introduce the basic ideas of probability and conditional probability and its dependence.
- Introduce discrete and continues random variable.
- Study simple application to reliability and life testing.

**Course Code: BA204**

**Course Title: Linear Algebra**

**Course Contents:**

- Learn the basic theory of linear algebra through Eigenvalues.
- Realize the wide applicability of linear algebra by examining applications.
- Learn some useful algorithms for linear systems

**Course Code: BA113**

**Course Title: Physics**

**Course Contents:**

- Understand the basic rules of **electrostatics** (Conductor and insulator) and Coulumb's law.
- Capacitors (Parallel plate capacitor, energy stored), and capacitors with dielectric between its plates.
- Electric currents in DC circuits, and RC circuits.
- Magnetism (force in a charge and current carrying conductor), generation of magnetic fields, electro-magnetic induction, magnetic flux, faraday's law. Lenz's law, and mutual induction-self induction.

Physical optic (huygen's principle of light-interference). Interference-thin films- Newton's rings.

**Course Code: EC134**

**Course Title: Fundamentals of Electricity and Electronics**

**Course Contents:**

- Understand the basic principles of Electricity and electronics
- Identify the electrical and electronic components and circuits.
- Understand the operation and uses of amplifiers and oscillators.
- Carry out simple electronic projects.

**Course Code: AR115**

**Course Title: Visual Studies**

**Course Contents:**

- Demonstrate understanding of the elements of interface visual design
- Use graphic vocabulary
- Use computer design compositions.
- Apply colour theories and principles of shapes in web designs

**Course Code: CS111**

**Course Title: Introduction to Computers**

**Course Contents:**

- Introducing Computational thinking
- Identifying the hardware and software components of a computer
- Introducing data types and understanding data internal representations
- Use a programming language to implement, test, and debug algorithms for solving simple problems.
- Introduce Modular programming
- Introducing the Internet and World Wide Web (if time allows)

**Course Code: CE216**

**Course Title: Digital Logic Design**

**Course Contents:**

- Know the basic differences between analog and digital systems
- Use binary numbers and codes
- Describe the operation of logic gates
- Apply Boolean Algebra on K-map
- Describing circuit operations using state diagrams
- Design a combinational and sequential logic circuits to simplify function

**Course Code: CE243**

**Course Title: Computer Architecture**

**Course Contents:**

- Present the various digital components used in the organization and design of digital computers.
- Explain the detailed steps that a designer must go through in order to design an elementary basic computer.
- Introduce the organization and architecture of the main units of a digital computer

**Course Code: CS143**

**Course Title: Introduction to Problem Solving and Programming**

**Course Contents:**

- Apply problem solving techniques.
- Discuss data types, file types, constants and variables.
- Use the various mathematical, logical and relational operators.
- Construct statements (assignments, expressions and functions).
- Perform program design techniques (flowcharts and pseudo-code).
- Use programming constructs (repetition, selection and sequencing)
- Discuss data structures (arrays and linked lists).
- Design, write, run and debug introductory programs using the currently adopted program language.

**Course Code: CS243**

**Course Title: Object-Oriented Programming**

**Course Contents:**

- Differentiate between the different editions of Java Language.
- Understand object oriented design principles.
- Understand the basic components of object oriented programming such as inheritance, abstraction, interfaces, inner classes and encapsulation.
- Understand the class members and access modifiers.
- Apply methods overloading and overriding.
- Handle runtime exceptions.
- Using Java API (java.lang, collection framework, ... etc.)
- Build user interfaces using Java swing package, and event driven programming.
- Understand object oriented relations (associations, aggregation, composition, inheritance and interface, .... etc).
- Create simple software system using Java.

**Course Code: CS244**

**Course Title: Advanced Programming Applications**

**Course Contents:**

- Build advanced user interfaces using Java swing package.
- Communicate with a DBMS using JDBC.
- Understand multi-threaded programming.
- Discuss distributed computing using RMI
- Understand the basics of Java web programming, such as servlets, JSPs.
- Understand Unix/Linux code development tools
- Understand scripting languages
- Work in teams and effectively communicate ideas and outcomes.

**Course Code: CS345**

**Course Title: Structure of Programming Languages**

**Course Contents:**

- Be familiar with several language paradigms and how they relate to different application domains.
- Understand the design space of programming languages, including concepts and constructs from past languages as well as those that may be used in the future.
- Understanding of the programming language we use by being able to identify and compare the same concept as it appears in different languages.
- Evaluating of programming models to provide a range of possible solutions and the ability to select the most optimized and relevant to the problem in hand.
- Understand the concepts and theory behind the implementation of high level programming languages
- Know significant details about a number of important techniques commonly used in compilers construction.

**Course Code: CS433**

**Course Title: Web Programming**

**Course Contents:**

- Write HTML files without using web generation tools.
- Specify the difference between client-side and server-side web programming.
- Master the basic syntax of JavaScript as a client-side scripting language.
- perceptual, dynamic and interaction aspects of virtual environments.
- Develop dynamic web pages using JavaScript
- Describe how server-side scripts work
- Install and administer an Apache Web Server in a UNIX environment.
- Understand what open-source software is.
- Understand what a server-side HTML-embedded scripting language is.
- Know the syntax of PHP.
- Use PHP to develop dynamic web sites
- Access a MySQL Database from a PHP script.

**Course Code: CS343**

**Course Title: Game Programming**

**Course Contents:**

- Understand the design space of 2D game programming, techniques including concepts and constructs as well as those that may be used in the future
- Understand principles of game design that make for a playable experience.
- Evaluating of Allegro game library to provide a wealth of up to date information pertaining to Allegro programming and gaming
- Participation from Allegro developers and the general public
- Know significant details about a number of important game techniques commonly used in the PC computer.
- Experiment with game design techniques and develop original games for PCs
- Write C/C++ game applications using Allegro game programming library

**Course Code: CS212**

**Course Title: Data Structures and Algorithms**

**Course Contents:**

- Revise concepts of Arrays, Structs, and Pointers
- Understand the concept of Linked List
- Understand & use Stacks
- Understand and implement Queues
- Understand the concept of Hashing
- Trees ( BST, AVL trees, Heaps,..)
- Graphs implementation and processing
- Searching and Sorting techniques

**Course Code: CS312**

**Course Title: Computing Algorithms**

**Course Contents:**

- Apply asymptotic time complexity analysis to choose among competing algorithms.
- Construct and solve recurrence equations describing the asymptotic time complexity given algorithm
- Implement efficient sorting algorithms
- Implement graph and tree structures
- Identify similarities and difference between breadth-first and depth-first search techniques
- Identify the use of dynamic programming techniques in algorithmic design.
- Describe how to write a parallel algorithms

**Course Code: CS311**

**Course Title: Theory of Computation**

**Course Contents:**

- Understand the capabilities and limitation of computational models
- Prove whether or not a given language is regular.
- Prove whether or not a given language is context-free.
- Design variants of Turing machines
- Understand the relationship between the regular, context-free and recursively enumerable languages.



**Course Code: CS202**

**Course Title: Discrete Structures**

**Course Contents:**

- Use logic to determine the validity of an argument.
- Construct the proof of a theorem.
- Understand the terminology, operations, and symbols of set theory.
- Use combinatorial techniques when needed in solving problems.
- Identify a function; specifically, surjective, injective, and bijective functions.
- Identify a relation; specifically, a partial order, equivalence relation, or total order.

**Course Code: CS301**

**Course Title: Numerical Methods**

**Course Contents:**

- Help student to work with symbolic representation as if they were concrete objects.(logical symbols)
- Help student to think abstractly.
- Help student to avoid logical errors.
- Help student to reason from definitions.
- Help student to use both direct and indirect argument to derive new results.
- Help student to think recursively by assuming similar problems of smaller nature that have been already solved and figuring out how to the larger problem.
- Help student to deal with discrete structures ,categorizing and revealing the underlying relations among them .(e.g. number theory, functions, relations, finite state automata)
- Help student to study the measurements of laws concerning combinations, probability.
- Help student to design an algorithm and determine whether it is correct or not (Mathematical induction)
- Show the students the extraordinary practical utility of some mathematical ideas.

**Course Code: IS373**

**Course Title: Database systems**

**Course Contents:**

- Explain database design concepts.
- Design a database system for a real-world problem.
- Implement and verify a database system using ORACLE.

**Course Code: IS474**

**Course Title: Advanced Database Systems**

**Course Contents:**

- Explain the concepts for modeling, designing, querying and managing large databases.
- Experiment with the modeling and design of distributed databases, data warehousing, web databases, XML databases, and mobile databases.

**Course Code: CE231**

**Course Title: Introduction to Networks**

**Course Contents:**

- Understanding the motivation that derived the research work in computer networks and overviewing the different contributions, and early research efforts that shaped the evolution of the Internet e research work in computer networks and overviewing the different contributions, and early research efforts that shaped the evolution of the Internet
- Introducing link multiplexing techniques (TDM, FDM, Statistical Multiplexing)
- Understanding the different Network Service Models (Packet switching vs Circuit switching)
- Introducing network delays and their effect on the network performance
- Understanding Network layer services and differentiating between the network core and edge
- Understanding routing and forwarding in the light of the two main classes of routing algorithms used over the Internet; Link State and Distance Vector routing protocols
- Introducing the Internet Protocol (IP), IP subnetting and Internet Control Message protocol (ICMP)
- Understanding the ISP tier structure and hierarchical addressing
- Understanding and differentiating between Flow control and congestion control and studying their implementation in the TCP/IP stack

**Course Code: CS321**

**Course Title: Systems Programming**

**Course Contents:**

- Build the architecture of a hypothetical machine, its assembly language, macro language.
- Program in assembly language.
- Build the structure and design of assemblers, linkers and loaders.
- Understand the concepts and theory behind the implementation of high-level programming languages through building a compiler/interpreter for a simple high-level language.

**Course Code: CS322**

**Course Title: Operating Systems**

**Course Contents:**

- Operating Systems overview
- Process Description & Control
- Threads
- Concurrency
- Deadlocks
- Memory Management
- Virtual Memory Management
- Uniprocessor, multiprocessor, and Real-time scheduling
- I/O management and Disk scheduling

**Course Code: CS427**

**Course Title: Embedded Systems Programming**

**Course Contents:**

- Identify and appreciate the meaning of embedded systems applications.
- Differentiate between Hard, Soft, and Firm real-time systems
- Use Microcontroller software with high level programming to develop task oriented applications in real time.
- Use real time operating system functions and tools for multi-tasking, inter-networking, memory limitations, hardware resource allocation, etc...
- Define the basic hardware components needed to implement Embedded systems.

**Course Code: CS366**

**Course Title: Introduction to Artificial Intelligence**

**Course Contents:**

1. Understand the basic concepts of symbolic artificial intelligence.
2. Understand different Methods for knowledge representation: state space, production rules, logic, etc..., and associated algorithms: search, forward/ backward inference algorithms, resolution, etc...
3. Implement a variety of intelligent applications using C, Prolog, and shells.
4. Use machine learning as a new paradigm for building intelligent systems.

**Course Code: IS461**

**Course Title: Decision Support Systems**

**Course Contents:**

- Understand the concepts of decision support systems structure and the principles of their design.
- Analyze typical decision situations to determine whether it is practical to support them with computer technology and, if so, how.
- Design and implement a decision support system.
- Understand emerging technologies and issues in the context of decision support systems as well as the management of information technology.
- Understand decision support, expert, and group decision support system use, development, and evolution
- 

**Course Code: CS452**

**Course Title: Computer Graphics**

**Course Contents:**

- Fundamental Concepts: analogue signals to discrete samples, raster vs. vector, pixels, GUI A image compression, color, graphics primitives, flicker-fusion, animation
- Rendering Techniques: graphics pipeline, modeling, 2D graphics, 3D graphics, transformation texture mapping, sampling, and ant aliasing
- Geometric Modeling: implicit and parametric forms, fractal images, transformation, Clipping

- Visualization: Apply Lighting & Texture mapping, behavior and interaction techniques, Recognize a variety of applications of visualization including representations of scientific, medical, and mathematical data; flow visualization; and spatial analysis.
- Animation: Use Alpha GV, timing, Implement interpolation methods for producing in-between positions and orientations, morphing techniques, LOD,
- Billboard
- PDC: Understand GPU architecture and apply simple GPU rendering CUDA code
- Project: Develop a computer graphics project to apply course concepts.

**Course Code:** CS451

**Course Title:** Human Computer Interaction

**Course Contents:**

- Apply a variety of interaction design processes and techniques and know when they are appropriate
- Examine an interactive product and explain what is good and bad about it in terms of the concepts, goals, and principles of interaction design
- Apply usability evaluation methods and know when they are appropriate
- Understand the role of social dynamics in interaction and how it applies in design, including concerns such as privacy, power, and accessibility
- Understand cognitive factors that affect usability
- Judge the availability and feasibility of different devices for interacting

**Course Code:** CS421

**Course Title:** Computer System Security

**Course Contents:**

- Identify threats to computer systems
- Outline security attacks and countermeasures
- Master classical and modern encryption techniques
- Experiment with authentication protocols
- Outline application layer security (E-mail and Web Security)
- Experiment with system security (Firewalls and Intrusion Detection)
- Outline main components of a security policy

**Course Code:** CS405

**Course Title:** System Modeling & Simulation

**Course Contents:**

- Understand the basic principles of the field of Modeling and Simulation.
- Master the modeling and performance evaluation of queuing systems using Markov chains
- Explore simulation definitions, history, background and languages.
- Highlight the different stages in building a discrete event simulation model and the architecture of simulation software.
- Study some applications as computer architecture, operating systems, and network for performance analysis.

**Course Code: IS171**

**Course Title: Introduction to Information systems**

**Course Contents:**

- Identify and explain the different types of Information Systems.
- Appreciate the value of information systems in modern business management and operations.
- Identify and describe different types of SDLC methodologies.
- State the difference between the logical and physical design of a system.
- Define the term organization and identify its components.
- Define data management concepts and terms.
- Analyse and design a solution for a “real-world” Computer Business Information System.
- Discuss the ethical issues involved in developing and interacting with Information Systems.

**Course Code: IS372**

**Course Title: Information Systems Theory & Practice**

**Course Contents:**

- Understand the role of information system in modern business and the competitive advantage they can deliver for a company
- Evaluate critically those current methods and techniques used to develop business and information strategies
- Secure and evaluate key aspects of IS/IT provision
- Understand the web impact in most business aspects and required web-based technologies
- Analysis of business and their processes to recognize where the application of information systems and other IT systems can improve the efficiency of those processes
- Organizing and implementing MIS
- Project Management for Information Systems
- Analysis of complex systems
- Group working methods
- Impact of change on stable systems

**Course Code: IS391**

**Course Title: Systems Analysis & Design**

**Course Contents:**

- Build a solid understanding of the information systems design and implementation phases of systems development life cycle (SDLC).
- Introduce several tools and techniques used in information systems design.
- Emphasize the use of systematic and structured methodologies in the design and implementation phases of SDLC.

**Course Code: SE291**

**Course Title: Introduction to Software Engineering**

**Course Contents:**

- Know the software life cycle.
- Understand how to analyse, design, implement and test large-scale software.
- Practice teamwork in developing software project.
- Introduce techniques and standard documents used in each stage of the software life cycle
- Work within a standard development process.
- Document process and product measurements.
- Utilize personal design and code reviews effectively.
- Analyze process data and propose improvements.

**Course Code: SE391**

**Course Title: Project Management**

**Course Contents:**

- Plan, analyze, develop, test, and implement an information system, emphasizing the phases most needed for the particular project .
- Work closely with the sponsors and users of the information system .
- Prepare a project notebook and web site to document all work on the project.
- Develop skills in using more software tools such as Microsoft Project , the Web, etc
- Prepare a personal portfolio demonstrating his/her ability to work in the IT field .
- Develop technical writing, teamwork, and presentation skills .

**Course Code: SE392**

**Course Title: Software Requirements & Specifications**

**Course Contents:**

- Revise software development cycle .
- Analyzing the problem.
- Understanding user and stakeholder needs .
- Defining the system .
- Constructing structural modal (Class diagram).
- Constructing dynamic model (Use-cases and sequence diagram).
- Requirement validation and checking .
- From Use Cases to implementation
- Tracing requirements
- Agile requirements methods

**Course Code: SE393**

**Course Title: Principle of Software Architecture**

**Course Contents:**

- Understand principles of software architecture and their application to the software development process
- Understand and be able to apply a variety of architectural styles
- Review and evaluate software architectures
- Specify a software architecture and implement a software system embodying it
- Use computer-aided software engineering (CASE) tools in an architecture-driven design process
- Work effectively as a member of a small team
- Communicate architecture and design issues in an oral presentation

**Course Code: SE491**

**Course Title: Software Component Design**

**Course Contents:**

- Understand basic components of object oriented programming
- Understand and apply object-oriented design patterns
- Understand the use of UML in the design process
- Design and implement small SW components
- Use computer-aided SW engineering (CASE) tools
- Perform independent research on software design
- Communicate SW design concepts in a brief oral presentation

**Course Code: SE492**

**Course Title: Software Verification**

**Course Contents:**

- Differentiate between the different s/w testing
- Understand s/w Validation and verification concepts.
- Understand different types of s/w testing.
- Understand the automatic of static analysis of s/w.
- Understanding test cases generations, and fundamentals.
- Understanding of s/w testing tools and measure matrices.
- Understanding of test design techniques (Black Box and white Box).
- Understanding of automation techniques for testing processes.
- Understand control flow graph testing.
- Understanding of statement, decision, and branch coverage.
- Understanding of Testing Management Activities
- Understanding of Performance Evaluation – Generic Models

**Course Code: SE493**

**Course Title: Software Quality Assurance**

**Course Contents:**

- Identify the unique characteristics and environment of SQA.
- Identify the various causes of software errors.
- Explain the need for comprehensive software quality requirements documents.
- Explain the SQA architecture that contains the components of SQA system.
- Discuss the importance of carrying out a contract review.

- Identify the elements of a quality plan.
- Identify the major software risk items.
- Compare the major review methodologies.
- Describe the various types of testing strategies and implementations.
- Compare automated testing and manual testing.
- List the contributions of CASE tools to product quality.
- Describe the importance of infrastructure component of SQA system.

**Course Code: SE494**

**Course Title: Formal methods in Software Engineering**

**Course Contents:**

- Understand the mathematical basis of formal methods.
- Understand the risks of poorly specified software systems
- Use formal specification languages in requirement specifications.
- Apply formal reasoning and proof techniques in software verification and validation.
- Understand the role of formal methods in software engineering practice.
- Model software systems using formal methods.
- Communicate software requirements and designs using a formal notation.

**Course Code: SE495**

**Course Title: Security in Software Engineering**

**Course Contents:**

- Understand software security best practices.
- Understand risk management frameworks and processes.
- Carry out code review using static analysis tools.
- Understand and use architectural risk analysis.
- Carry out penetration testing.
- Carry out security testing.
- Understand abuse case development.
- Acquire threat modeling techniques.

**Course Code: SE496**

**Course Title: Software Engineering Process**

**Course Contents:**

- Understand software development best practices.
- Apply RUP model.
- Use iterative approach for software development.
- Design S/W using component-based architectures.
- Integrate and use cases in RUP.
- Acquire skills in RUP's project management.
- Apply RUP's business modeling.
- Apply RUP's requirements discipline.
- Implement analysis, design, implementation, and testing disciplines within RUP.
- Understand the configuration and change management disciplines.



**Course Code:** NC272

**Course Title:** Fundamentals of Business

**Course Contents:**

- Conduct business in the global economy.
- Start and Expand a small business
- Manage a business, information systems, and production
- Manage employees.
- Develop marketing strategies to satisfy customers.
- Manage accounting and financial resources.

**Course Code:** NC252

**Course Title:** Principles of Marketing

**Course Contents:**

- Apply marketing theory and concepts toward solving managerial marketing problems.
- Analyze in written and oral presentation form, basic marketing principles related to specific business situations (perform critical thinking.)
- Understand and identify marketing opportunities, and how marketing concepts and tools can be applied to achieve organizational objectives.
- Appreciate the role of marketing executives within a complex matrix of business and society issues

**Course Code:** NC264

**Course Title:** Principle of Microeconomics

**Course Contents:**

- Demonstrate an understanding, usage and application of basic economic principles
- Describe and apply the methods for analyzing consumer behavior through demand and supply, elasticity and marginal utility
- Understand the role of alternative property rights in resource allocation
- Identify and appraise various models of how markets are organized, and the price and output decisions for maximizing profit
- Know how markets that fail to use resources efficiently create unintended effects
- Strengthen problem solving skills by applying economic criteria to business decisions, international trade and public policy

**Course Code:** NC273

**Course Title:** Global Business

**Course Contents:**

- Introduction to global business affecting IT and software development .
- Topics of interest include globalization and the international business environment, differences between countries and their cultures, virtual global organisations, international product life cycle, international strategic management, international logistics management, quality and process management, project management and international human resources management.

**Course Code: NC282**

**Course Title: Financial Accounting**

**Course Contents:**

- Understand the underlying concepts of accounting including the accounting cycle, accrual accounting, accounting for current assets, fixed assets and current liabilities.
- Understand the accounting process and prepare the financial statements for a small business.
- Be knowledgeable users of financial statements.
- Have an introductory-level understanding of computerized accounting systems and how to use those systems.

**Course Code: CS481**

**Course Title: Computers & Society**

**Course Contents:**

- Understand the ethical, social, legal, and professional issues surrounding the emerging information technologies.
- Sharpen his skill in moral reasoning, imagination, and behavior.
- Develop a coherent set of positions on the issues in this course, and a plan for implementing them in his professional life.

## 7. Program admission requirements

The students are admitted to CCIT on the basis of their general secondary school certificate record (or equivalent).

Applicants must not be more than 22 years of age at the time of enrolment. A student must be in sound health and free from diseases, and a recent medical report to this effect should be presented.

Applicants should satisfy the conditions and scores stated by the Supreme Council of Egyptian Universities, these are:

- 1- High school certificates "Thanaweya Amma" Mathematics Section.
- 2- High school certificates "Thanaweya Amma" Science Section, with additional preparatory courses to cover mathematics background needed for studying in CCIT. (These courses will be considered as zero credit).
- 3- IGCSE, American Diploma, or equivalent diplomas from any other country

Transfer applications from accredited universities and academies are also accepted by the Academy.

The Academy considers readmission applicants for students who were unable to continue their studies for one semester or more in light of their studies before their withdrawal

## 8. Regulations for progression and program completion:

According to study program at the college, all CCIT students study the same courses in the first four semesters after which they declare one of the three majors for the next four semesters.

In order to complete the program of all departments and acquire a Bachelor's degree in it, the student must successfully achieve 144 credit hours in four years at least.

Regular attendance in all classes is required. Students are expected to arrive on time and remain in class for the entire period scheduled. The responsibility for work missed due to any type of absence rests with the student. Attendance is mandatory

Students are expected to attend every class session. Each instructor will maintain current records and will, upon request, furnish an accurate report of any student's attendance pattern to the Registration Office. Attendance is checked from the first class meeting; therefore, late registrants will have some absences when they first meet a class.

When students are absent, for any reason, they are still responsible for the content of the missed lecture and for any assignments. If the absence causes the student to miss a major examination: that examination may be made up if the instructor is convinced that the absence was justified. A student's attendance record is a legitimate part of the criteria, which the instructor may use in assigning a grade for the courses. It is the responsibility of students to be aware of the attendance policies for their classes.

After the student has missed over 15% of class, the student will be asked to withdraw from the course and a grade of "W" will be entered. In case of illness or other forcing circumstances, absence may reach 20% without forcing withdrawal.

Students withdrawing from study during a semester must fill out a withdrawal permit; have it properly executed by the Registrar in order to leave a clear record. A student completing this process before the beginning of the final examination period will have a "W" record on the permanent record.

A student who withdraws without following this required procedure will be awarded an "F". Students cannot officially withdraw from study after the last day of classes prior to final examinations or while under disciplinary investigation.

Students are personally responsible for completing all requirements established for their degree by the College / Department. It is the student's responsibility to know the requirements for the appropriate degree program.

Faculty advisors will assist students in preparing schedules, completing degree plans, and generally will counsel students on academic matters, but advisors may not assume responsibility for the student's academic progress and ultimate success or failure in a given program of study.

Any substitution, waiver, or exemption for any established requirement or academic standard may be accomplished only with the approval of the department Head and the College Dean.

Constant follow ups are done for students with special needs after each assessment to evaluate their academic level of achievement. The academic advisor of each student is in charge of solving any problems that she/he may encounter.

### Graduation Requirements

To be qualified for Bachelor of Science the candidate must:

- Complete 144 credit hours of approved work
- Earn a cumulative grade point average of at least 2.00
- Complete practical training

## 9. Evaluation of Program ILOs

Method	Evaluated ILOS
Written exams including short Quizzes	Knowledge and understanding
Oral Exams including those taken during laboratory hours	Knowledge and understanding
Assignments and course term work	Intellectual Skills
Mini projects (single student or team work)	Professional, Practical Skills, General and Transferable Skills
Practical Work	Professional, Practical Skills, General and Transferable Skills
Graduation Projects (team work)	Professional, Practical Skills, General and Transferable Skills

## 10. Methods of Program evaluation:

	Evaluator	Method
1	Students	Questionnaires, surveys, checklists
2	Alumni	Questionnaires, surveys, checklists
3	Employers	Individual interviews, Questionnaires, Meetings
4	External auditors and Examiners	Documentation review, Observation
5	Instructors	Focus groups, Course surveys, Meetings

### Head of Department

**Name: Prof. Dr. Samah Ahmed Senbel**

**Sign:**

**Date:**    /    /