Department of Computer Science

Chairperson:	El-Hajj, Wassim M.
Professor:	Safa, Haidar H.; Turkiyyah, George M.
Associate Professors:	Abu Salem, Fatima K.; Attie, Paul C.; El-Hajj, Wassim
Assistant Professors:	Bou Ammar, Haitham; Dhaini, Ahmad; Elbassuoni, Shady; Jaber, Mohamad
Visiting Assistant Professor:	Nassar, Mohamed
Senior Lecturer:	Jureidini, Wadi' N.
Instructors:	Bdeir, Mahmoud; Sidani-Bohsali, Hayat A.
Part-time Instructors:	Afra, Lama; Al Feel, Roaa; Aoude, Loa; Asfour, Yousif; Assaf, Rida; Farran, Rami; Hamam, Mustafa; Hamandi, Ahmad; Moubarak, Mohammad; Sakr, Jad; Sobh, Amine; Yassine, Zahraa; Zaatari, Ayman

The Department of Computer Science offers a program leading to the degree of Bachelor of Science (BS) in Computer Science. It also offers a program leading to the degree of Master of Science (MS) in Computer Science. For more information about the department visit http://www.cs.aub.edu.lb/.

BS in Computer Science Mission Statement

The department of Computer Science prepares students for advanced study and professional careers in the dynamically changing world of computing and information technology. The BS program aims to produce graduates with a solid foundation in computing at both the theoretical and practical levels and who have the ability to design, build, and deploy sophisticated systems using current technologies in a broad array of areas. It also develops an appreciation of the transformative impact that computing has had on a wide variety of disciplines. Students are trained in quantitative reasoning, the use of fundamental principles and ideas (abstraction, modularity, data structures, algorithmics, computability, calculus, logic) for analysis and problem solving, and disciplined development of modern software systems. The department has vigorous research programs in graphics and multimedia, networking for security, high-performance computing, data mining and information retrieval, and software engineering and is committed to providing opportunities for students to get engaged in research in these areas.

Degree Requirements

To graduate with a BS in computer science, a student must complete the following requirements:

University General Education Requirements

- The General Education requirements are the following:
- English Communication Skills (6 credits), Arabic Communication Skills (3 credits)
- Humanities (12 credits), Social Sciences (6 credits), Natural Sciences (6 credits)
- Quantitative Thought (3 credits)

Major Requirements

- Computer science: CMPS 200, CMPS 205, CMPS 212, CMPS 213, CMPS 253, CMPS 255, CMPS 256, CMPS 257, CMPS 258, CMPS 272, CMPS 277, CMPS 299, and 9 additional credits in computer science courses numbered 230 and above.
- Technical electives: four credits to be chosen as follows:
  - 3 cr. course selected from the following list: CMPS elective numbered 230 or above, BIOL 251, PHYS 228, PHYS 235, MATH 261, STAT 234, STAT 238, ENTM 220, ENTM 230, and INDE 410
  - 1 cr. course selected from the available one credit CMPS special topic courses (CMPS 297 - Special Topics in Computer Science)
- Mathematics: MATH 201, MATH 211 (or CMPS 211), MATH 218 (or 219), STAT 230 (or 233).

All prospective computer science majors are expected to complete CMPS 200, CMPS 205, MATH 201, MATH 211 or CMPS 211, and CMPS 212 in the sophomore year. Computer science majors are expected to complete CMPS 213, CMPS 253, CMPS 255, CMPS 256, CMPS 257, and CMPS 258 in the junior year and maintain an average grade of at least 70 in computer science courses. Students must have an average of 70 or more in CMPS 200 and CMPS 212 before they are allowed to enroll in CMPS courses numbered 230 and above.

A minor in computer science requires 18 credits: CMPS 200, CMPS 211, CMPS 212, CMPS 256 and 6 additional credits in computer science courses (CMPS) numbered 230 or above. A minimum of 9 credits must be taken in the department.**[Note: This minor is not open to EECE students.]**

Sample Study Plan

A typical study plan could have the following distribution of CMPS courses:

First Year

- First Semester: CMPS 200, CMPS 211
- Second Semester: CMPS 205, CMPS 212, CMPS elective

Second Year

- First Semester: CMPS 213, CMPS 255, CMPS 256
- Second Semester: CMPS 253, CMPS 258, CMPS 277

Third Year

- First Semester: CMPS 257, CMPS 272, CMPS elective
- Second Semester: CMPS 299, CMPS elective

CMPS 101 Introduction to Computer Science 2.2; 3 cr.
This course introduces the skills, concepts, and capabilities needed for effective use of information technology (IT). It includes logical reasoning, organization of information, managing complexity, operations of computers and networks, digital representation of information, security principles, and the use of contemporary applications such as effective Web search, spreadsheets, and database systems. Also it includes a basic introduction to programming and problem solving through scripting web applications. *Every semester.*

CMPS 200 Introduction to Programming 3.3; 3 cr.
An introduction to a disciplined approach to computer programming and problem solving, utilizing a block-structured high level language, with an emphasis on procedural abstraction and good programming style. This course covers the basic repetition and selection constructs, procedures and functions, parameter passing, and scope of variables. *Every semester.*

CMPS 205 Introduction to Computing Systems 1.2; 1 cr.
This course provides a broad introduction to computer science. It is meant to expose students to some of the ideas of the field as well to develop fluency in the use of information technology. The course introduces operations of computers and networks, World Wide Web and standards, systems for representing and organizing information, management of complexity, security principles and algorithmic thinking. *Corequisite: CMPS 212 Annually.*

CMPS 206 Computers and Programming for the Arts 2.2; 3 cr.
This course is an introductory computer course that presents computing and information, and illustrates their use. The student is introduced to computers and their role in society with emphasis on conceptual understanding as well as operational proficiency. Topics include principles of computer operations both from the hardware and software perspectives, basic networking concepts, web authoring concepts including HTML, cascading style sheets, and publishing, and data manipulation using spreadsheets and databases. This course is meant to be a computer literacy course open to Arts students only. *No credit is given to computer science majors. Students can get credit for only one of CMPS 206 or CMPS 209. Annually.*

CMPS 207 Programming for Digital Art 3.3; 3 cr.
This course introduces students to the technical and conceptual skills necessary for developing web sites and for analyzing and visualizing real data . In web design, students will learn HTML5 and CSS3. In data analysis and visualization, students will learn to code using Python with an emphasis on organizing, analyzing, and plotting data. Visualizations produced by Python can then be embedded into html pages. The core skills learned in this course will be applicable to most programming languages. *Not open to computer science students. Annually.*

CMPS 209 Computers and Programming for the Sciences 2.2; 3 cr.
This course is designed to cover the essential computer skills needed by students in sciences and business majors. The course introduces how computers and Internet technologies work by emphasizing conceptual understanding as well as practical operational proficiency. Specifically, the course covers the essential concepts needed for designing spreadsheet applications, building personal relational databases, and programming using Visual Basic. Other topics covered include networking basics. *No credit is given for computer science majors. Students can get credit for only one of CMPS 206 or CMPS 209. Every semester.*

CMPS 211 Discrete Structures 3.0; 3 cr.
This course introduces students to discrete structures, focusing on those relevant to computing sciences. Topics covered include Logic and Proofs, Sets, Sequences, Functions, Growth of Functions, Algorithms and their complexities, Induction and Recursion, Counting, and Recurrence Relations. *This course is equivalent to MATH 211. Annually.*

CMPS 212 Intermediate Programming with Data Structures 3.3; 3 cr.
A continuation of CMPS 200, this course consolidates algorithm design and programming techniques, emphasizing large programs. This course also provides a detailed study of data structures and data abstraction, and an introduction to complexity considerations and program verification. *Prerequisite: CMPS 200. Corequisite: CMPS 205. Every semester.*

CMPS 213 C/C++programming 1.2; 1 cr.
This course exposes students to the C and C++ programming languages. The course covers basic syntax, defining structures and classes, I/O, pointers, arrays, memory management, references, overloading, templates, the Standard Template Library, inheritance and polymorphism. *Prerequisite: CMPS 212. Corequisite: CMPS 255. Annually.*

CMPS 230 Digital Media Programming 3.0; 3 cr.
The class is an introduction to digital media programming and processing. The course explains the essential technology behind images, animations, sound, and video and how to write interactive programs that manipulate these media in creative ways. The class assumes basic knowledge in Java or a first course in programming. *Prerequisite: CMPS 200.*

CMPS 251 Numerical Computing 3.1; 3 cr.
Techniques of numerical analysis: number representations and round-off errors, root finding, approximation of functions, integration, solving initial value problems, Monte-Carlo methods. Implementations and analysis of the algorithms are stressed. Projects using MATLAB or a similar tool are assigned. *Prerequisites: CMPS 200 or EECE 230, and MATH 201. This course is equivalent to MATH 251. Annually.*

CMPS 253 Software Engineering 3.0; 3 cr.
A course that introduces the fundamentals of software engineering, with emphasis on the requirements elicitation and specification, and analysis and design phases of the software life cycle. Specifications are given as a set of operations (with pre- and post-conditions), and using a generic data model, and the design as a module dependency diagram where both data and procedural decomposition are emphasized. The course also introduces verification and testing of a design with respect to its specification, and the use of modularity and decomposition to ensure tractability of the verification. Students will apply the concepts learned to develop a software system. *Prerequisite: CMPS 212. Annually.*

CMPS 255 Computer Architecture 3.0; 3 cr.
A structured overview of the fundamentals of designing digital computer systems. Topics covered include digital logic and systems, machine level representation of data, assembly level machine organization, memory system organization and architecture, CPU implementation and virtual machines, and exposure to one or more micro/mini architectures. *Prerequisites: CMPS 211 and CMPS 212. Corequisite: CMPS 213. Annually.*

CMPS 256 Algorithms and Data Structures 3.0; 3 cr.
A systematic study of algorithms and advanced data structures, and their complexity. Topics include techniques for designing efficient computer algorithms, proving their correctness, and analyzing their complexity; as well as advanced searching, sorting, selection, priority queues, binary search trees, graph, hash tables, and matrix algorithms. *Prerequisites: CMPS 211 and CMPS 212. Annually.*

CMPS 257 Theory of Computation 3.0; 3 cr.
A course that covers basics of Automata and Language Theory, Computation Theory, and Complexity Theory. Topics include regular expressions, finite automata, context-free grammars and parsing, pushdown automata, closure properties, Turing machines, Church's thesis, reductions and decidability, time complexity and NP-completeness, space complexity, polynomial-space and log-space computations, circuit complexity, probabilistic computations and complexity classes, approximation algorithms, and selected topics as time permits. *Prerequisites: CMPS 211 and CMPS 212. Annually.*

CMPS 258 Programming Languages 3.0; 3 cr.
A course on the principles and programming styles that govern the design and implementation of contemporary programming languages, a history and overview of programming languages, fundamental issues in language design, and an introduction to language translation. This course focuses on design issues in imperative, object-oriented, functional, and rule-based paradigms. Various programming languages will be studied and experimented with to illustrate key concepts. *Prerequisite: CMPS 212. Annually.*

CMPS 272 Operating Systems 3.0; 3 cr.
This course provides an introduction to the fundamentals of operating system function, design, and implementation. It contains a theory component about the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation. The course divides into three major parts. The first part of the course discusses concurrency (processes, threads, scheduling, synchronization, and deadlocks). The second part of the course discusses memory management (memory management strategies and virtual memory management). The third part of the course concerns file systems, including topics such as secondary storage systems and I/O systems. If time permits, the following topics will be briefly examined: virtualization, security, distributed synchronization, and perhaps other topics. A case study of a contemporary operating system like UNIX accompanies the course. *Prerequisites: CMPS 213, CMPS 255 and CMPS 256. Annually.*

CMPS 273 Systems and Network Programming 3.0; 3 cr.
This course focuses on the programming aspects of networking protocols. Topics include: designing and building programming applications that use computer networks, fundamental concepts required to build iterative and concurrent client/server networking applications using sockets. Then it moves to explain low level networking programming and other advanced socket topics. The course also presents the emerging peer-to-peer computing along with some tools needed to develop P2P applications. *Prerequisite: CMPS 272. Annually.*

CMPS 274 Compiler Construction 3.0; 3 cr.
A course that covers syntax specifications of programming languages, parsing theory, top-down and bottom-up parsing, parser generators, syntax-directed code generation, symbol table organization and management, dynamic storage allocation, code optimization, dataflow analysis, and register allocation. *Prerequisites: CMPS 255, CMPS 258 and CMPS 257. Biennially.*

CMPS 277 Database Systems 3.0; 3 cr.
An overview of the nature and purposes of database systems and an introduction to data modeling: entity relationship model, relational model with relational algebra, relational calculus and SQL; integrity constraints; file organization and index files; normalization. *Prerequisite: CMPS 256. Annually.*

CMPS 278 Web Programming and Design 3.0; 3 cr.
This course introduces the fundamentals needed to program on the Internet, and the state of the art technologies used in designing and developing rich multi-tiered web based applications. It presents the basics of client-side/server-side web programming and the skills and tools needed to create dynamic Web-based applications. It provides in-depth coverage of various markup languages (XHTML, Dynamic HTML and XML) and their associated cascading style sheets, several client side and server side scripting languages (JavaScript, PHP) in addition to AJAX-enabled rich Internet applications, client-side technologies, web services, Web Servers, and multi-tiered applications using relational database systems. *Prerequisite: CMPS 200. Annually.*

CMPS 281 Numerical Linear Algebra 3.0; 3 cr.
A course on direct and interactive methods for solving general and special systems of linear equations, covering LU decomposition, Choleski decomposition, nested dissection, marching algorithms; Jacobi, Gauss-Seidel, successive over-relaxation, alternating directions, and conjugate gradient iterative methods. This course is equivalent to MATH 281. *Prerequisites: MATH 218 or 219; and MATH 251 or CMPS 211. Annually.*

CMPS 282 Advanced Software Engineering 3.0; 3 cr.
A course on state of the art software engineering for large distributed and concurrent systems. Fundamental principles and concepts for specifying, designing, analyzing, implementing, and testing such systems. Concurrent object oriented paradigms. Design patterns. Use of tools. Documentation using both formal and informal descriptions. Students will develop at least one large software system as part of the course. *Prerequisite: CMPS 253. Annually.*

CMPS 284 Computer Networks 3.0; 3 cr.
An introduction to network architectures and protocols, placing emphasis on Internet design principles and methodology. Specific topics include application layer protocols, network programming, transport protocols, circuit switching and packet switching, routing algorithms, multicast, local and wide area networks, error detection and correction, and performance evaluation. *Prerequisite: CMPS 256. Annually.*

CMPS 285 Computer Graphics 3.0; 3 cr.
A course that covers the practice of, and underlying mathematical foundation for, interactive graphics programming. Topics include basic graphics systems, graphics primitives and attributes, windows and viewports, clipping, geometric transformations, color systems, 2D texture mapping, and introduction to 3D graphics. Programming in OpenGL will be used. *Prerequisite: CMPS 212. Annually.*

CMPS 286 Computer-Aided Geometric Design 3.0; 3 cr.
A course that discusses the representation of free-form curves and surfaces in modeling objects by computers, including curve approximation and interpolation, spline curves (Bezier and B-splines), visual smoothness of curves, geometric continuity, parameterization of curves, introduction to surface interpolation and approximation, and spline surfaces (Bezier and B-splines). *Prerequisite: CMPS 212. Biennially.*

CMPS 287 Artificial Intelligence 3.0; 3 cr.
An introduction to the principles and techniques that enable computers to behave intelligently. This course covers basic problem solving methods, knowledge representation, reasoning methods, learning from samples and from experience, expert systems and knowledge acquisition, machine learning, and neural networks. Several projects are given, some of which are in Prolog. *Prerequisites: CMPS 256 and CMPS 258. Annually.*

CMPS 288 Internals of Database Management Systems 3.0; 3 cr.
A course on the internals of database management systems, especially relational DBMS. Topics include query processing and optimization, transaction processing, concurrency control, recovery, distributed transactions, database security, client-server, multi-tier architectures, and web deployed database systems. *Prerequisite: CMPS 277. Annually.*

CMPS 289 Human Computer Interaction 3.0; 3 cr.
This course describes the psychological principles of human-computer interaction. Evaluation of user interfaces. Usability engineering. Task analysis, user-centered design, and prototyping. Conceptual models and metaphors. Software design rationale. Design of windows, menus, and commands. Voice and natural language I/O. Response time and feedback. Color, icons, and sound. Internationalization and localization. User interface architectures and APIs. Case studies and project. *Prerequisites: CMPS 230 and CMPS 253. Biennially.*

CMPS 296 Computer Science Tutorial 1–3 cr.
Prerequisite: Senior standing.

CMPS 297 Special Topics in Computer Science 1–3 cr.
A course on selected topics which change according to the interests of the instructors and/or students. Topics are chosen from state-of-the-art innovations in software and computer information systems. *Prerequisite: Consent of instructor. Annually.*

CMPS 299 Software Graduation Project 3 cr.
A course to enhance students' skills with practical experience giving them the opportunity to integrate knowledge accumulated in different courses. In this course, students must deliver a software product, which passes through the design, analysis, implementation, testing, and evaluation stages. *Prerequisites: CMPS 253, CMPS 272, CMPS 277, and senior standing. Annually.*

41 Credits in Computer Science

Modes of Analysis	English and Arabic (9)	Humanities (12)	Social Sciences (6)	Natural Sciences (6)	Technical Electives (4)	Quantitative Thought (32+9+12)
Lecture Courses (9+12+6+6+4+53)	• Required Arabic courses (3): ARAB 201 A or B, or any upper level course (3), as determined by placement • Required English courses (usu¬ally 6): ENGL 203(3), 204(3), as determined by placement	• Required credits in the humanities: 12 credits includ¬ing 6 credits from CVSP	• Required Courses (6)	• Required natural[1] science courses (6)	• Required 3 credit course (3): CMPS elective numbered 230 or above, BIOL 251, PHYS 228, PHYS 235, MATH 261, STAT 234, STAT 238, ENTM 220, ENTM 230, or INDE 410 • Required 1 credit elective(1): CMPS 297(1)	• Required CMPS courses (32): CMPS 200(3)+205(1), 212(3)+213 (1), 253(3), 255(3), 256(3), 257(3), 258(3), 272(3) 277(3), 299(3) • Required CMPS electives (9): to be chosen from CMPS courses above 230 • Required mathemat¬ics courses (12): MATH 201(3), MATH 211(3) (or CMPS 211), MATH 218 (or 219), STAT 230 (or 233)
Seminar (0)						
Laboratory				PHYS 228L		CMPS 200, 205, 212, 213, 278
Research Project (0)						

[1] Natural science courses are numbered 200 and above and drawn from biology, chemistry, geology or physics, open to science students.