QFO-AP-FI-MO02	Course Syllabus :اسم النموذج	جامعة فيلادلفيا Philadelphia University
رقم الاصدار : 1 (Revision)	الجهة المصدرة:كلية تكنولوجيا المعلومات	
التاريخ: 2017/11/05	الجهة المدققة: عمادة التطوير والجودة	
عدد صفحات النموذج:		

Course Syllabus	
Course Title: Algorithms and Advanced Data Structures	**Course code: 750724**
Course Level: ***Master***	**Course prerequisite(s) and/or corequisite(s): None**
Lecture Time:	**Credit hours: 3**

Academic Staff Specifics

Name	Rank	Office Location	Office Hours	E-mail Address

Course Description

This course is intended for students already accepted to study the MSc in Computer Science in the Faculty of Information Technology. Students can expect to gain from the course the skills and techniques required to undertake problem solving at MSc level. It aims to emphases the concepts of analysis and design of algorithms and measure their complexity through different strategies. The practical aspect should be emphasized.

Course Objectives

This course aims to offer advanced skills in Problem Solving (Conventional problem solving process, software engineering problems solving process. Algorithm design patterns, framework, and instance. Algorithm design pattern for and with reuse), Algorithms Quality Assessment (Complexity assessment, Maintainability assessment, Correctness assessment), Conventional Algorithms Design Patterns overview (Divide & Conquer, Greedy, Dynamic Programming, Graph Algorithms, Algebraic simplification and transformation, Parallel and distributed, Randomized, Probabilistic, Heuristic and approximate), and Advanced Algorithm Design Patterns (Formal Specification, Genetic, fault tolerance)

Course Components

Textbooks

1. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani,. Algorithms. McGraw-Hill Higher Education, 2007
2. Kleiberg, J. & Tardos, E. Algorithm Design. Addison Wesley, 2006

In addition to the above, the students will be provided with handouts by the lecturer.

Teaching Methods

Duration: 15 weeks, 45 hours in total. Lectures: 15 hours, 1 per week. Tutorial (case study in classroom): 21 hours, 2 per week. Seminar: 3 (15 mn at the end of each lecture). Laboratories: 15 hours in total, 1-hour per week (free lab). Exams: 6 hours (3h for the mid and 3 for final exam). The last week is reserved to practical works examination.

Learning Outcomes

- ***Knowledge and understanding***
  - be prepared for some of the demands of, and skills required for, work in IT and IT-related industries
  - having been introduced to the skills and knowledge necessary to Problem solving

- ***Cognitive skills (thinking and analysis)***
  - Understand A wide range of principles, methodologies, and tools available to the Algorithm developer. All these direction informed by research.
  - Understand the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality.
  - Understand the application of computing in a business context

- ***Communication skills (personal and academic)***
  - Have skills and knowledge necessary to undertake the project
  - Be able to display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities.

- ***Practical and subject specific skills (Transferable Skills)***
  - Solve a wide range of problems related to the Algorithm design, Analysis, development, and use.
  - Design, analysis and implementation of different kind of algorithms.
  - Plan and undertake a major individual project, and prepare and deliver coherent and structured verbal and written technical report.

Learning Outcomes Achievement (http://www.philadelphia.edu.jo/program-specification-cs-dept**)**

Developed: A1, A2, A3, A4, A5, B1, B3, B4, C1, C2, C3, C4, C5, C6. D1, D2, D3. D4, D5, D6

Assessed: A2, A4, B1, B3, C1, C3, C4, C5, C6

Assessment Instruments

Allocation of Marks	
Assessment Instruments	**Mark**
Midterm examination	**30%**
Final Exam (written unseen exam)	**40 %**
Reports, research projects	**30%**
Total	**100%**

** Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.*

Practical Submissions

The assignments that have work to be assessed will be given to the students in separate documents including the due date and appropriate reading material.

Documentation and Academic Honesty

Submit your homework covered with a sheet containing your name, number, course title and number, and type and number of the home work (e.g. assignment, and project).

Any completed homework must be handed in the class on the due date. After the deadline "zero" will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:

- A brief report to explain your findings.
- Your solution of given problem

For the research report, you are required to write a report similar to a scientific research paper. It should include:

- *Abstract:* It describes the main synopsis of your paper.
- *Introduction:* It provides background information necessary to understand the research and getting readers interested in your subject. The introduction is where you put your problem definition, summary of contribution, related work, and is likely where the bulk of your sources will appear.
- *Methods (Algorithms and Implementation):* Describe your methods here. Summarize the algorithms (if any) generally, highlight features relevant to your project, and refer readers to your references for further details. Information from sources must be rephrased in own words, "copy-and-paste" from documents, found for example on the Internet, is NOT allowed. It is allowed to use short quotations, or figures, from other documents, but then the source MUST be clearly stated in the reference list (please check copy rights). Papers not fulfilling these rules will be failed.

- *Results and Discussion (Benchmarking and Analysis):* This section is the most important part of your paper. It is here that you demonstrate the work you have accomplished on this project and explain its significance. The quality of your analysis will impact your final grade more than any other component on the paper. You should therefore plan to spend the bulk of your project time not just gathering data, but determining what it ultimately means and deciding how best to showcase these findings.
- *Conclusion:* The conclusion should give your reader the points to "take home" from your paper. It should state clearly what your results demonstrate about the problem you were tackling in the paper. It should also generalize your findings, putting them into a useful context that can be built upon. All generalizations should be supported by your data, however; the discussion should prove these points, so that when the reader gets to the conclusion, the statements are logical and seem self-evident.
- *Bibliography:* Refer to any reference that you used in your assignment. Citations in the body of the paper should refer to a bibliography at the end of the paper.

- **Protection by Copyright**

1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

- **Avoiding Plagiarism**

1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/Module Academic Calendar

Week	Basic and support material to be covered	HW
(1)	**Part 1/ Problem Solving** *Chapter 1. Introduction/A Software Engineering Process for Problems Solving* Conventional problem solving process, A software engineering problems solving process. Algorithm design patterns, framework, and instance. Algorithm design pattern for/with reuse. *Tutorials based on relevant research works papers*	PW on Design patterns
(2)	**Part 2/ Algorithms Quality Assessment** *Chapter 2. Complexity assessment/* - Mathematical background overview, Asymptotic Notations (O, Ω, θ), applications - Lower bound theory, NP hard and NP Complete Problems	
(3)	*Chapter 3. Maintainability assessment /* - Error avoidance properties: (weaknesses and strengths: structured style and idiom, Modular, Object-Oriented paradigm, Reuse). Tutorials *Chapter 4. Correctness assessment /* - Algorithm testing and proving fundamentals and applications *Tutorials based on relevant research works papers*	PW on structured programming
(4)	**Part 3/ Conventional approaches Design Patterns** *Chapter 5. Conventional Algorithms Design Patterns overview /* - Divide & Conquer (Pattern, Frameworks, instances: Binary search, Merge sort, …).	PW on D&C RW on Quality and

	Tutorials based on relevant research works papers	D&C
(5)	- Greedy (Pattern, Frameworks, instances: Marketing optimization problems …). *Tutorials based on relevant research works papers*	PW on Greedy
(6)	- Dynamic Programming (Pattern, Frameworks, instances: Marketing optimization problems). *Tutorials based on relevant research works papers*	PW on Dynamic
(7)	- Graph Algorithms (Basic search and traversal algorithms, Backtracking, Branch & Bound). *Tutorials based on relevant research works papers*	PW on BT & BB
(8)	**Mid Exam**	
(9)	- Parallel and distributed (Pram, Mesh, Hypercube). *Tutorials based on relevant research works papers*	PW on Par/Distr. Algorithms
(10)	- Algebraic simplification and transformation (functional, logic, ..), Randomized. Probabilistic - Heuristic and approximate *Tutorials based on relevant research works papers*	RW on Greedy, Dynamic, BB, BT, Parallel/Distributed Alg
(11)	**Part 4/ Advanced approaches** *Chapter 6 Advanced Algorithm Design Patterns* - Formal Specification	
(12)	- Fault tolerance	
(13)	Tutorials	
(14)	**Homework Exam**	
(15)	**Written Final Exam**	

Expected workload

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module References (sample)

Students will be expected to give the same attention to these references as given to the Module textbook(s):

1. S. Vaidyanathan (Author). Data structures, algorithms and applications in C++. New Delhi: CBC Publishers & Distributors Pvt Ltd 2013
2. Michel Raynal (Author). Concurrent programming: algorithms, principles, and foundations. Heidelberg: Springer 2013
3. Anany Levitin (Author). Introduction to the design and analysis of algorithms New Delhi: Dorling Kindersley/Pearson 2013 2nd ed.
4. Chandra Mohan (Author). Design and analysis of algorithms. New Delhi: PHI Learning Private Limited 2012 2nd ed.
5. Narasimha Karumanchi (Author). Data structures and algorithms : made easy in Java Edison, NJ: CareerMonk Publications 2012 2nd ed
6. Clifford A. Shaffer (Author). Data structures & algorithm analysis in Java Mineo;a, New York: Dover Publications 2011 3rd ed.
7. Anany Levitin (Author). Introduction to the design and analysis of algorithms Boston: Pearson/Addison - Wesley 2007 2nd ed.
8. Simon Harris (Author) James Ross (Author). Beginning algorithms. Indianapolis, Indiana: Wiley Publishing, Inc. 2006
9. Mark Allen Weiss (Author). Boston: Pearson Addison Wesley 2006 3rd ed.
10. Michael T. Goodrich (Author) Roberto Tamassia (Author). Data structures and algorithms in Java Hoboken, NJ: John Wiley & Sons 2006 4th ed.
11. Jon Kleinberg (Author) Eva Tardos (Author). Algorithm design Boston: Pearson addison Wesley 2006

Research papers (provided in the course along with course chapters in tutorials)

Website(s):

http://ecourse.philadelphia.edu.jo/login/index.php // Prof. Said
http://www.cs.berkeley.edu/~vazirani/algorithms.html // e- textbook
http://www.cs.yale.edu/homes/avi/db-book/db5/index.html
http://db.cs.sfu.ca/sections/publication/dood/dood.html