| | | |
|---|---|---|
| **Identification** | Title | *Data Structure and Algorithms* |
| | Department | *Computer Science & Engineering* |
| | Program | *B.Sc. - 3 credits* |
| | Semester | *Spring 2018* |
| | Instructor | *Seyed Amir Hossein Siahpooshha (PhD)* |
| | E-mail: | siahpooshha@gmail.com |
| | Classroom/hours | *Neftchilar Campus, room#* |
| | Pre-requisites | *Excellent knowledge of programming* |
| | Language | *English (intermediate level)* |
| | Type | *Major (Compulsory)* |
| **Course Resources** | *1. Introduction to Algorithms, 3rd Edition, (2009) by Thomas H. Cormen, Charles E. Leiserson,*<br>*2. Ronald L. Rivest, Clifford Stein, The MIT Press, Cambridge, Massachusetts.*<br>*3. Class notes* | |
| **Course Outline** | *Software engineering is the study of ways in which to create large and complex computer applications and that generally involve many programmers and designers. At the heart of software engineering is with the overall design of the applications and on the creation of a design that is based on the needs and requirements of end users. While software engineering involves the full life cycle of a software project, is includes many different components - specification, requirements gathering, design, verification, coding, testing, quality assurance, user acceptance testing, production, and ongoing maintenance.*<br>*Having an in-depth understanding on every component of software engineering is not mandatory, however, it is important to understand that the subject of data structures and algorithms is concerned with the coding phase. The use of data structures and algorithms is the nuts-and-blots used by programmers to store and manipulate data.*<br>*This course focuses on the essentials of data structures and algorithms. Attempts will be made to understand how they work, which structure or algorithm is best in a particular situation in an easy to understand environment.* | |
| **Course Objective** | *The main objective of this course as follow:*<br>*The topics covered will be similar to those found in introductory algorithms and data structures courses in computer science departments across the world: sorting and searching algorithms, categorising efficiency in time and memory use, linked list and tree data structures, hash tables, stacks and queues. The objectives are that students should know something of all of these by the end of the course. As well as knowing about them, students should be familiar enough with the concepts that should they need to take any of them further and make use of them, students will be able to do so.* | |
| **Learning Outcomes** | *Design, analysis and implementation of efficient and secure software and applications.* | |
| **Course Policy** | *Lesson organization: General information on the subject will be provided for the students during lectures.*<br>*Attendance: Participation of students at class is important. Students should inform dean's office about missing lessons for particular reasons (illness, family issues and etc.).*<br>*Lates: Students have to get to class on time.*<br>*Exams: All the issues related to the participation and admission to the exam are regulated by the faculty dean. Topics of midterm and final exams are provided for the students before the exams.*<br>*Violation of the rules of the exams: Cheating and plagiarism is equal to Zero.*<br>*The rule for completing the course: In accordance with the University rules the overall success rate to complete the course should be 60% or above. The students who failed the exam would* | |

| | | |
|---|---|---|
| | *be to take this subject next semester or next year.*<br>*Rules of conduct for Students: Disruption of the lesson and not following ethical norms during the lesson, as well as conduction of the discussions by the students without permission and using mobile phones is forbidden.* | |
| **Evaluation** | Midterm Exam | *30%* |
| | Attendance | *5%* |
| | Project | *10%* |
| | Homework (3) | *15%* |
| | Activity | *10%* |
| | Final Exam | *30%* |

| **Weekly Lectures** | |
|---|---|
| 1 | Overview of Data Structure and Algorithms |
| 2 | Elementary Data Structures:<br>Stacks and queues<br>Linked lists<br>Implementing pointers and objects<br>Representing rooted trees |
| 3 | Growth of Functions:<br>Asymptotic notation<br>Standard notations and common functions |
| 4 | Getting Started:<br>Insertion sort<br>Analyzing algorithms<br>Designing algorithms |
| 5 | Divide-and-Conquer:<br>The maximum-subarray problem<br>Strassen's algorithm for matrix multiplication<br>The substitution method for solving recurrences<br>The recursion-tree method for solving recurrences<br>The master method for solving recurrences<br>Proof of the master theorem |
| 6 | Midterm Exam |
| 7 | Heapsort:<br>Heaps<br>Maintaining the heap property<br>Building a heap<br>The heapsort algorithm<br>Priority queues |
| 8 | Quicksort:<br>Description of quicksort<br>Performance of quicksort<br>A randomized version of quicksort<br>Analysis of quicksort |
| 9 | Binary Search Trees:<br>What is a binary search tree?<br>Querying a binary search tree |

| | |
|---|---|
| | Insertion and deletion |
| 10 | Sorting in Linear Time:<br>Lower bounds for sorting<br>Counting sort<br>Radix sort<br>Bucket sort |
| 11 | Hash Tables:<br>Direct-address tables<br>Hash tables |
| 12 | Hash Tables:<br>Hash functions |
| 13 | Hash Tables:<br>Open addressing |
| 14 | Red-Black Trees:<br>Properties of red-black trees<br>Rotations |
| 15 | Red-Black Trees:<br>Insertion<br>Deletion |
| 16 | Final Exam - |