|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | **ECTS Course Description Form** | | | | | | | | | | | | | | | | |
|  | **PART I (Senate Approval)** | | | | | | | | | | | | | | | | | | | | |
|  | **Offering School** | **Engineering** | | | | | | | | | | | | | | | | | | | |
|  | **Offering Department** | **Computer Engineering** | | | | | | | | | | | | | | | | | | | |
|  | **Program(s) Offered to** | **Computer Engineering** | | | | | | | | | | | **Compulsory** | | | | | | | | |
|  |  | | | | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | | | |  | | | | | | | | |
|  | **Course Code** | **CS311** | | | | | | | | | | | | | | | | | | | |
|  | **Course Name** | **Algorithms** | | | | | | | | | | | | | | | | | | | |
|  | **Language of Instruction** | **English** | | | | | | | | | | | | | | | | | | | |
|  | **Type of Course** | **Compulsory - Lectures** | | | | | | | | | | | | | | | | | | | |
|  | **Level of Course** | **Undergraduate** | | | | | | | | | | | | | | | | | | | |
|  | **Hours per Week** | **Lecture: 3** | | | | **Laboratory:** | | | **Recitation:** | | **Practical:** | | | | **Studio:** | | | **Other:** | | | |
|  | **ECTS Credit** | **7** | | | | | | | | | | | | | | | | | | | |
|  | **Grading Mode** | **Letter Grade** | | | | | | | | | | | | | | | | | | | |
|  | **Pre-requisites** | **CS102 and CS213** | | | | | | | | | | | | | | | | | | | |
|  | **Co-requisites** |  | | | | | | | | | | | | | | | | | | | |
|  | **Registration Restriction** |  | | | | | | | | | | | | | | | | | | | |
|  | **Educational Objective** | **The main objective of this course is to provide the students with a knowledge on foundations of problem solving, computational efficiency, and experience in the design and implementation of algorithms commonly employed in computer science and computational problems.** | | | | | | | | | | | | | | | | | | | |
|  | **Course Description** | **Introduction to the main concepts of design and analysis of algorithms. Overview of basic analysis techniques: approximating functions asymptotically, bounding sums, and solving recurrences. Discussion of efficiently solvable problems with a focus on design techniques such as divide-and-conquer, randomization, dynamic programming, amortization, and greedy algorithms. Illustration of various new concepts through algorithms applied to problems related to sets, sequences, strings, graphs etc.** | | | | | | | | | | | | | | | | | | | |
|  | **Learning Outcomes** | **LO1: Employ recursion as a problem solving and programming technique** | | | | | | | | | | | | | | | | | | |  |
|  | **LO2: Design algorithms employing randomization, dynamic programming, greedy heuristics** | | | | | | | | | | | | | | | | | | |
|  | **LO3: Analyze runtime efficiency of an algorithm.** | | | | | | | | | | | | | | | | | | |
|  | **LO4: Interpret and analyze algorithmic solutions to problems related to sets, sequences, strings, graphs** | | | | | | | | | | | | | | | | | | |
|  | **LO5: Engineer and implement algorithms (200-500 lines of code)** | | | | | | | | | | | | | | | | | | |
|  | **LO6: Design experiments for algorithm correctness and efficiency testing** | | | | | | | | | | | | | | | | | | |
|  | **PART II ( Faculty Board Approval)** | | | | | | | | | | | | | | | | | | | | |
|  | **Basic Outcomes (University-wide)** | | **No.** | **Program Outcomes** | | | | | | | | **LO1** | | **LO2** | | **LO3** | **LO4** | | **LO5** | **LO6** | |
|  | **PO1** | **Ability** to communicate effectively and write and present a report in Turkish and English. | | | | | | | | 1 1 1 1 1 1  1 1 1 1 1 1  2 2 2 2 2 2  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  3 3 1 3 3 3  0 0 0 0 0 3  3 3 3 3 3 3  0 0 0 0 0 0  3 3 3 3 1 1  0 0 0 0 3 3  2 2 2 2 1 1 | | | | | | | | | |
|  | **PO2** | **Ability** to work individually, and in intra-disciplinary and multi-disciplinary teams. | | | | | | | |
|  | **PO3** | **Recognition** of the need for life-long learning and **ability** to access information, follow developments in science and technology, and continually reinvent oneself. | | | | | | | |
|  | **PO4** | **Knowledge** of project management, risk management, innovation and change management, entrepreneurship, and sustainable development. | | | | | | | |
|  | **PO5** | **Awareness** of sectors and **ability** to prepare a business plan. | | | | | | | |
|  | **Faculty/ Program Specific Outcomes** | | **PO6** | **Understanding** of professional and ethical responsibility and **demonstrating** ethical behavior. | | | | | | | |
|  | **PO7** | **Ability** to define complex engineering  problems, develop models and  implement solutions for these  problems | | | | | | | |
|  | **PO8** | **Ability** to conduct lab experiments by using  computers and the ability of collecting, analyzing and interpreting data. | | | | | | | |
|  | **PO9** | **Ability** to apply the knowledge of  mathematics, science and engineering  principles to solve problems in computer  engineering. | | | | | | | |
|  | **PO10** | An **understanding** of current contemporary  issues and impact of engineering solutions  in legal and ethical levels | | | | | | | |
|  | **PO11** | **Ability** to understand and apply discrete  mathematics concepts. | | | | | | | |
|  | **PO12** | **Ability** to use modern engineering  techniques, tools and information  technologies and develop software  equipment and software. | | | | | | | |
|  | **PO13** | **Ability** to analyze, design and manage the  hardware/software computer system  requirements with limited resources and  conditions by modern engineering  principles. | | | | | | | |
| **PART III (Department Board Approval)** | | | | | | | | | | | | | | | | | | | | | |
| **Course Subjects, Contribution of Course Subjects to Learning Outcomes, and Methods for Assessing Learning of Course Subjects** | | | **Subjects** | **Week** | | |  | | | | | **LO1** | | **LO2** | | **LO3** | **LO4** | | **LO5** | **LO6** | |
| **S1** | 1 | | | General overview of the course, role of algorithms in computer science | | | | | A1/3 | |  | |  |  | |  |  | |
| **S2** | 2 | | | Inductive design with an example:  Insertion sort, analyzing algorithms, Divide-and-conquer with an example:  Merge sort, analysis of merge sort | | | | | A1/3 | |  | | A1/3 | A1/3 | |  |  | |
| **S3** | 3 | | | Asymptotic notation, common functions, solving recurrences, common recurrences | | | | |  | |  | | A1/3 |  | |  |  | |
| **S4** | 4 | | | Randomized algorithms with an  example: Quicksort, expected runtime analysis. | | | | | A1/3 | | A1/3 | | A1/3 | A1/3 | |  |  | |
| **S5** | 5 | | | Heapsort, sorting in linear time and counting sort, Information theoretic lower bounds | | | | |  | |  | | A1/3 | A1/3 | |  |  | |
| **S6** | 6 | | | Asymptotic Analysis, Find/union data structure | | | | |  | |  | | A1/3 |  | |  |  | |
| **S7** | 7 | | | Dynamic programming: Longest  common subsequence problem, Knapsack problem, pseudo-polynomial algorithms | | | | | A1/3 | | A1/3 | | A1/3 | A1/3 | |  |  | |
| **S8** | 8 | | | Dynamic programming: Optimal  binary search tree problem | | | | | A1/3 | | A1/3 | | A1/3 |  | |  |  | |
| **S9** | 9 | | | Greedy algorithms: Activity selection, Huffman encoding | | | | | A1/3 | | A1/3 | | A1/3 |  | |  |  | |
| **S10** | 10 | | | Introduction to Graphs, Representing graphs, Graph traversals via DFS, BFS | | | | | A1/3 | |  | | A1/3 | A1/3 | |  |  | |
| **S11** | 11 | | | Graph decompositions: connected, biconnected, strongly connected components, cycle finding | | | | | A1/3 | |  | | A1/3 | A1/3 | |  |  | |
| **S12** | 12 | | | Minimum Spanning trees, Kruskal's, Prim's Algorithms, Single Source shortest paths, Dijkstra's Algorithm | | | | | A1/3 | |  | | A1/3 | A1/3 | |  |  | |
| **S13** | 13 | | | All-pairs shortest paths, flow networks, max-flow min-cut | | | | | A1/3 | |  | | A1/3 | A1/3 | |  |  | |
| **S14** | 14 | | | Project demos  Overall evaluation of the course | | | | |  | |  | |  |  | | A4 | A4 | |
| **Assessment Methods, Weight in Course Grade, Implementation and Make-Up Rules** | | | **No.** | **Type** | | | | **Weight** | | **Implementation Rule** | | | | **Make-Up Rule** | | | | | | | |
| **A1** | **Exam** | | | | *70* | | *There is 1 midterm*  *exam and 1 final exam for the course. The midterm’s weight is 20 and the final exam’s weight is 50. Exam dates will be shown on the tentative schedule and it can be changed according to the course schedule.* | | | | If a student misses an exam and  provides an acceptable legitimate  document, a make-up exam will be  provided. | | | | | | | |
| **A2** | **Quiz** | | | |  | |  | | | |  | | | | | | | |
| **A3** | **Homework** | | | | *10* | | *There are 2 homework, each with weight 5, for the course. Each student*  *should prepare his/her homework by himself, herself. Submissions should be neatly presented.* | | | | There will be no make-up for the  homework. | | | | | | | |
| **A4** | **Project** | | | | *20* | | *There is a coding project that involves engineering of a non-trivial algorithm, entailing 200-500 lines of code. Students should prepare their own test cases and data to test for correctness and efficiency. This is an individual project. At the end each student will make a demo.* | | | | There will be no make-up for the project. | | | | | | | |
| **A5** | **Report** | | | |  | | - | | | | - | | | | | | | |
| **A6** | **Presentation** | | | |  | | - | | | | - | | | | | | | |
| **A7** | **Attendance/ Interaction** | | | |  | | - | | | | - | | | | | | | |
| **A8** | **Class/Lab./**  **Field Work** | | | |  | | - | | | | - | | | | | | | |
| **A9** | **Other** | | | |  | |  | | | |  | | | | | | | |
| **TOTAL** | | | | | **100%** | | | | | | | | | | | | | |
| **Evidence of Achievement of Learning Outcomes** | | | Students will demonstrate learning outcomes through midterm exam, homework assignments, the final exam and the project. Every topic is tested with at least one exam or homework question. In order to pass, a student needs to accumulate certain percentage of points and this percentage is determined by the class mean. Students' overall understanding of algorithms, efficiency concepts are put to test in the coding project where they are expected to implement a non-trivial algorithm and design its experimentation and testing. | | | | | | | | | | | | | | | | | | |
| **Method for Determining Letter Grade** | | | The method on which the letter grade is based on will be announced at the beginning of the semester, and  this method may be subjected to change depending on the performance of the students. 2 midterms, 2  homework assignments, and the final exam are used for grading. Letter grades are tentatively determined  using the table below. Here “-x” means (average-3-x) and “+x” means (average+3+x), and each denotes  the minimum points necessary for the corresponding letter grade.   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Total points** | +25 | +20 | +15 | +10 | +5 | Class Average ± 3 | -5 | -10 | -15 | -20 | | **Letter Grade** | A | A- | B+ | B | B- | C+ | C | C- | D+ | D | | | | | | | | | | | | | | | | | | | |
| **Teaching Methods, Student Work Load** | | | **No** | **Method** | | | | **Explanation** | | | | | | | | | | **Hours** | | | |
| ***Time applied by instructor*** | | | | | | | | | | | | | | | | | | |
| **1** | **Lecture** | | | | Lecturing and utilizing white board. Sample questions and answers to strengthen learning. In class exams. | | | | | | | | | | 3x14=42 | | | |
| **2** | **Interactive Lecture** | | | |  | | | | | | | | | |  | | | |
| **3** | **Recitation** | | | |  | | | | | | | | | |  | | | |
| **4** | **Laboratory** | | | |  | | | | | | | | | |  | | | |
| **5** | **Practical** | | | |  | | | | | | | | | |  | | | |
| **6** | **Field Work** | | | |  | | | | | | | | | |  | | | |
| ***Time expected to be allocated by student*** | | | | | | | | | | | | | | | | | | |
| **7** | **Project** | | | | |Preliminary readings, coding, testing, demo | | | | | | | | | | 50 | | | |
| **8** | **Homework** | | | | Answers of given questions are prepared at home | | | | | | | | | | 2x8=16 | | | |
| **9** | **Pre-class Learning of Course Material** | | | | New subjects are learned by watching videos or reading course  notes before class. | | | | | | | | | | 5x14=70 | | | |
| **10** | **Review of Course Material** | | | | Review of the subjects before exams in order to prepare. | | | | | | | | | | 18 | | | |
| **11** | **Studio** | | | |  | | | | | | | | | |  | | | |
| **12** | **Office Hour** | | | | One office hour per week is allocated for students’ questions | | | | | | | | | | 14 | | | |
| **TOTAL** | | | | | 210 | | | | | | | | | | | | | |
| **IV. PART** | | | | | | | | | | | | | | | | | | | | | |
| **Instructor** | | | **Name** | | | | | Cesim Erten | | | | | | | | | | | | | |
| **E-mail** | | | | | cesim.erten@antalya.edu.tr | | | | | | | | | | | | | |
| **Phone Number** | | | | | *+90-242-2450000* | | | | | | | | | | | | | |
| **Office Number** | | | | | *A1-28* | | | | | | | | | | | | | |
| **Office Hours** | | | | | *TBA* | | | | | | | | | | | | | |
| **Course Materials** | | | **Mandatory** | | | | | *INTRODUCTION TO ALGORTHMS, CORMEN, LEISERSON, RIVEST,*  *STEIN, THE MIT PRESS, 2ND EDITION, 2001.* | | | | | | | | | | | | | |
| **Recommended** | | | | | *THE ART OF COMPUTER PROGRAMMING, D. KNUTH, ADDISON*  *WESLEY, 1998.* | | | | | | | | | | | | | |
| **Other** | | | **Scholastic Honesty** | | | | | Violations of scholastic honesty include, but are not limited to cheating, plagiarizing,  fabricating information or citations, facilitating acts of dishonesty by others, having  unauthorized possession of examinations, submitting work of another person or work  previously used without informing the instructor, or tampering with the academic work  of other students. Any for of scholastic dishonesty is a serious academic violation and  will result in a disciplinary action. | | | | | | | | | | | | | |
| **Students with Disabilities** | | | | | Reasonable accommodations will be made for students with verifiable disabilities. | | | | | | | | | | | | | |
| **Safety Issues** | | | | |  | | | | | | | | | | | | | |
| **Flexibility** | | | | | Circumstances may arise during the course that prevents the instructor from fulfilling  each and every component of this syllabus; therefore, the syllabus is subject to change.  Students will be notified prior to any changes. | | | | | | | | | | | | | |