**Course Description**

Based on **NCEAC.FORM.001-C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | CS3613 (CY) and CS2203 (CS) | | | |
| **Course Title** | Analysis of Algorithms | | | |
| **Credit Hours** | 3 | | | |
| **Prerequisites by Course(s) and Topics** | Data Structures | | | |
| **Assessment Instruments with Weights** (homework, quizzes, midterms, final, programming assignments, lab work, etc.) | * 4 Quizzes (Announced/Unannounced) 15% * 4 Assignments 15% * Digital Course/ Coursera 5% * 1 Project/Presentation 10% * 1 Mid Term 25% * Final Exam 30% | | | |
| **Course Coordinator** |  | | | |
| **URL (if any)** | http://vle.riphah.edu.pk/ | | | |
| **Current Catalog Description** |  | | | |
| **Textbook** (or **Laboratory Manual** for Laboratory Courses) | * Thomas H. Cormen et al. “Introduction to Algorithms” 3rd Edition. * Anany Levitin “Introduction to the design and analysis of algorithms” 3rd Edition. | | | |
| **Reference Material** | * Lecture Handouts * In-class Activities * Various Online Resources * Dasgupta, Papadimitriou, and Vazirani: Algorithms, McGraw-Hill, 2006 * Kleinberg, Tardos: Algorithm Design, Addison Wesley Longman, 2006. * Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos, * ACM Transactions on Algorithms | | | |
| **Course Goals** | **Goal:** Students should be able to Design and Analyze intermediate level  algorithms for correctness, completeness and discuss complexity trade-offs.  **Learning Objectives:** Students should be able to   1. Design algorithms using different algorithms design techniques i.e. Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms 2. Analyze the time and space complexity of different algorithms by using standard analysis techniques for recursive and non-recursive algorithms. 3. Discuss on Asymptotic notations, standard complexity classes and representation of time complexities in asymptotic notations of standard complexity functions 4. Describe, compare, analyze, and solve general algorithmic problem types: Sorting, Searching, String Processing, Graph. 5. Implement the algorithms, compare the implementations empirically, and apply fundamental algorithms knowledge to solve real-world problems. 6. Understanding of P, NP, NPC, NPH problems, NP-Completeness and Approximate Problems. | | | |
| **Topics Covered in the Course, with Number of Lectures on Each Topic** | *Please see details on the next page.* | | | |
| **Laboratory Projects/Experiments Done in the Course** | *Please see details on the next page.* | | | |
| **Programming Assignments Done in the Course** | *Please see details on the next page.* | | | |
| **Class Time Spent on** (in credit hours) | **Theory** | **Problem Analysis** | **Solution Design** | **Social and Ethical Issues** |
| 0.75 | 0.75 | 1.35 | 0.15 |
| **Oral and Written Communications** | Every student is required to submit at least one Project and one presentation. Presentation and project demonstration of typically 30 minutes. | | | |

|  |  |  |
| --- | --- | --- |
| **Code#** | **Bloom’s Taxonomy**  **And Course Learning Outcomes** | **Weightage** |
| **2.0** | **Understand** |
| 2.1 | Understanding of P, NP, NPC, NPH problems, NP-Completeness and Approximate Problems. | 10% |
| **3.0** | **Apply** |
| 3.1 | Implement the algorithms, compare the implementations empirically, and apply fundamental algorithms knowledge to solve real-world problems. | 20% |
| **4.0** | **Analyze** |  |
| 4.1 | Analyze the time and space complexity of different algorithms by using standard analysis techniques for recursive and non-recursive algorithms. | 15% |
| 4.2 | Discuss on Asymptotic notations, standard complexity classes and representation of time complexities in asymptotic notations of standard complexity functions | 15% |
| **5.0** | **Evaluate** |  |
| 5.1 | Describe, compare, analyze, and solve general algorithmic problem types: Sorting, Searching, String Processing, Graph. | 20% |
| **6.0** | **Create** |  |
| 6.1 | Design algorithms using different algorithms design techniques i.e. Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms | 20% |

**Topics Covered & Weekly Course Plan**

Based on **Bloom’s Taxonomy**

**[K]** Knowledge **[C]** Comprehension **[A]** Application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Time** | **Topic** | **B.T.** | **Course Objective** |
| 01 | 3 Hours | Course Policies / Guidelines, Introduction / Overview, Basics of Algorithms, Mathematical Foundation | K, C | c |
| 02 | 3 Hours | Growth of Function, Asymptotic Notations, Time and Space Complexity, Iterative Algorithms Analysis, Algorithms Constructs, Fibonacci, and Factorial Algorithms | K, C, A | b, c |
| 03 | 3 Hours | Sorting (Bubble, Selection, Insertion, Counting, Radix) | K, C | a, b, d, e |
| 04 | 3 Hours | Recursive Algorithms, Analysis of Recursive Algorithms, Recurrence Tree, Recurrence Relation | K | b, c |
| 05 | 3 Hours | Iterative Substitution, Telescoping Sum, Divide and Conquer Approach, Sorting (Merge Sort, Quick Sort), Fibonacci and Factorial Recursive Algorithms | K, C, A | b, c, d |
| 06 | 3 Hours | Analysis of Data Structures (Stack, Queue, Linked List, Hash Table, Binary Tree) | K, C, A | b, c, d, e |
| 07 | 3 Hours | Graph Theory, Graph Terminology, Representation of Graphs, Graph Traversal (BFS & DFS) | K | c, d |
| 08 | 3 Hours | **Midterm**, Midterm Solution |  |  |
| 09 | 3 Hours | Optimization Problems, Greedy Algorithms, Shortest Path Finding in Graph (Dijkstra’s Algorithm) | K, C | b, d |
| 10 | 3 Hours | Greedy Algorithms: Minimum Spanning Trees (Kruskal’s Algorithm, Prim’s Algorithm), Huffman Coding for Data Compression | K, C | b, c, d |
| 11 | 3 Hours | Dynamic Programming, Tabular Method, Knapsack Problem, | K, C | D |
| 12 | 3 Hours | Dynamic Programming: Matrix Multiplication, Matrix Chain Multiplication Problem | K, C | b, c, d |
| 13 | 3 Hours | String Matching, Longest Common Subsequences | K, C | D |
| 14 | 3 Hours | NP Complete Problems and Solutions using Approximation Algorithms | K, C | c, d, f |
| 15 | 3 Hours | Approximation Algorithms | K, C | c |
| 16 | 3 Hours | Revision, Project Presentations | K, C, A |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Map course LOs with the program LOs. (Place course LO #s in the left column and program LO #s across the top.)** | | | | | | | | | | | |
| **Course**  **LOs #** | **Program Learning Outcomes** | | | | | | | | | | |
| **PLO1** | **PLO2** | **PLO3** | **PLO4** | **PLO5** | **PLO6** | **PLO7** | **PLO8** | **PLO9** | **PLO10** | **PLO11** |
| **2.1** |  |  | **X** |  |  |  |  |  |  |  |  |
| **3.1** |  |  | **X** |  |  |  |  |  |  |  |  |
| **4.1** |  | **X** |  |  |  |  |  |  |  |  |  |
| **4.2** | **X** | **X** |  | **X** |  |  |  |  |  |  |  |
| **5.1** |  |  |  | **X** |  |  |  |  |  |  |  |
| **6.1** |  |  | **X** |  |  |  |  |  |  |  |  |