|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Syllabus** | **Curriculum** | **Deployment Strategy and**  **Tool** | **Cross-cutting issues**  **integrated** | **PO, PSO and CO** | **Attainment Verification** |
| 1. | DESIGN AND ANALYSIS OF ALGORITHMS | * Algorithm analysis is an important part of a broader computational complexity theory, which provides theoretical estimates for the resources needed by any algorithm which solves a given computational problem. These estimates provide an insight into reasonable directions of search for efficient algorithms. * Algorithms are used in every part of computer science. They form the field's backbone. In computer science, an algorithm gives the computer a specific set of instructions, which allows the computer to do everything, be it running a calculator or running a rocket. * When solving a problem, choosing the right approach is often the key to arriving at the best solution. In psychology, one of these problem-solving approaches is known as an algorithm. An algorithm is a defined set of step-by-step procedures that provides the correct answer to a particular problem. | 1. Chalk and   Talk method   1. PPT | * Business   Ethics   * Human   values | PO1:Engineering Knowledge  PO2:Problem Analysis  PO4:Conduct Investigations Of Complex Problems |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  | PSO2:Problem Solving Skill |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  | CO1:Understand the basic concepts of algorithms. Analyse worst-case, best case and average case running times of recursive and non-recursive algorithms using asymptotic notations. |
|  |  |  |
|  |  | CO2: Describe computational solution to well-known problems like searching, sorting etc. and understand different fundamental data structures. |
|  |  |  |
|  |  | CO3: Describe the divide-and-conquer and decrease and conquer model and explain when an algorithmic design situation calls for it. Derive and solve recurrences describing the performance of divide-and-conquer algorithms |
|  |  |  |
|  |  |  |
|  |  | CO4:Analyse the greedy method and transform and conquer method and explain when an algorithmic design situation calls for it. Study various greedy algorithms, and analyse the performance of each one of them. |
|  |  | CO5:Analyse the dynamic-programming method and explain when an algorithmic design situation calls for it. Design dynamic-programming algorithms, and analyse each one of them |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | CO6:Understand the basic concepts of Backtracking, Branch and Bound, P, NP, NP-complete, NP-hard. Also analyse the algorithms under these concepts. |

