**Design and Analysis of Algorithms**

**Course Objectives :**

Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

**COURSE OUTLINE:**

**1. What is an algorithm?**

* Notation for programs
* Proof techniques
* Basics review: Sets - Functions - Limits - Simple series

**2. Fundamentals**

* Instances and problems - Elementary operations.
* Efficiency
* Average and worst-case analysis
* Examples

**3. Asymptotic notation**

* Introduction
* A notation for "the order of"
* The omega notation
* The theta notation
* The conditional asymptotic notation

**4. Analysis of algorithms**

* Analyzing control structures
* Using a barometer
* Amortized analysis
* Solving recurrences

**5. Data structures**

* Arrays, stacks and queues
* Records and pointers
* Lists, graphs, trees and associative tables
* Heaps o Disjoint set structures

**6. Greedy algorithms**

* Making change
* General characteristics of Greedy algorithms
* Graphs MST - Kruskal's and Prims's algorithms
* Graphs: shortest paths
* Knapsack problem o Scheduling

**7. Divide - and - Conquer**

* Multiplying large integers
* Binary search o Sorting by: merging and quicksort
* Finding the median
* Matrix multiplication
* Exponentiation
* Quick look at cryptography

**8. Dynamic programming**

* Making change
* Principles of optimality
* The knapsack problem
* Shortest paths - Floyd's algorithm
* Chained matrix multiplication

**9. Introduction to probabilistic algorithms - Parallel algorithms**

**10. Introduction to computational complexity**