**ABET Course Syllabus – CS3112**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | CS3112 | **Credits** | 3 |
| **Title** | Analysis of Algorithms | **Coordinator** | Behzad Parviz |

**Course Information**

1. **Catalog Description:** Methods for the design and analysis of correct and efficient computer algorithms; applications to classical problems of searching, sorting, graph optimization and combinatorial optimization. Graded ABC/NC.
2. **Prerequisites**: CS2013 and CS2148
3. **Contact Hours**: Lecture 2 hours, laboratory 3 hour.
4. **Required/Elective:** This course is required in the BS program.

**Textbook**

Introduction to Algorithms (3rd Edition). By: Cormen, Leiserson, Rivest and Stein

**Course Goals**

The Student Learning Outcomes that are addressed by the course are:

* SLO1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.
* SLO 3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.

Other outcomes of instruction:

At the end of the course, students are able to

* Have the required Mathematical Foundation to calculate the Computational Complexity of Algorithms.
* Understand the Fundamentals of the Analysis of Algorithm Efficiency.
* Choose between Exact and Approximate Problem Solving.
* Prove an Algorithm’s correctness.
* Analyze an algorithm.
* Have knowledge of important problem types such as: Sorting, Searching, String Processing, Graph Problems, Combinatorial Problems
* Understand and use Asymptotic Notations and Basic Efficiency Classes.
* Perform Mathematical Analysis of Recursive and Non-recursive Algorithms.
* Understand the Space and Time Trade-Offs.

**Topics Covered**

* Mathematical Foundations
  + Summation Formulas
  + Logarithms
  + Induction
  + Lower and Upper bounds
  + Asymptotic Notation
  + Recurrence Relations
  + Master Theorem
  + Loop Invariants.
* Analysis of the Correctness and of the Computational Complexity of Computer Algorithms.
* Growth of Functions
* Divide-and-Conquer
* Probability analysis
* Randomized algorithms
* Advanced Data Structures
  + Binary Search Trees
  + Balanced Trees
  + Heaps
  + Indirect Heaps
  + Priority Queues
  + Dictionaries
  + Hash Tables.
* Graph Algorithms and Searching and Sorting Algorithms.
  + Binary search tree
  + Red-black tree
  + Minimum spanning tree
  + Breadth-first search
  + Depth-first search
* Design Techniques