

**COURSE OUTLINE**

**Course Name:** Design and Analysis of Algorithms

**Course Code**: HCE214

**Lecturer:**  Mr. Zvakafa

**Contact Details:** Cell: 0774322606

E-mail:szvakafa@ceic.uz.ac.zw

**Prerequisites:** This course is intended for students who have programming experience and the course data structures and Algorithms.

# Overview:

This core course covers good principles of algorithm design, elementary analysis of algorithms. The emphasis is on choosing appropriate algorithms and designing correct and efficient algorithms to solve problems.

# Learning outcomes:

In this course students will:

* learn good principles of algorithm design;
* learn how to analyse algorithms and estimate their worst-case and average-case behaviour (in easy cases);
* Learn how to apply their theoretical knowledge in practice (via the practical component of the course).
* Argue the correctness of algorithms using inductive proofs and invariants.
* Analyse worst-case running times of algorithms using asymptotic analysis.

# Course Content

1. Introduction to Design and analysis of algorithms
2. What is an algorithm?
3. Fundamental of the Algorithmic Problem Solving
4. Important Problem Types
5. Program cost and asymptotic analysis
6. The analysis framework and Growth of Functions
7. Asymptotic Notations and Basic Efficiency Classes
8. Mathematical Analysis of Recursive Algorithms
9. Substitution method
10. Iteration method
11. Brute Force / Exhaustive Search / Decrease-and-Conquer
12. Selection Sort ,Bubble Sort, Insertion Sort
13. Sequential Search and Brute-Force String Matching
14. Worst case analysis of merge sort, quick sort and binary search
15. Depth first search
16. Breadth first search
17. Design and analysis of Divide and Conquer Algorithms (Divide and Conquer algorithm design strategy)
18. Recursive Algorithm Analysis
19. Merge Sort
20. Quick Sort
21. Dynamic programming
22. Introduction to the DP Paradigm
23. Development of a Recursive Algorithm
24. Elimination of Redundant Work from a Recursive Algorithm
25. Transform into a memorized recursive algorithm
26. Transform into an iterative algorithm
27. Optimized Algorithm
28. Development of a recursive definition to solve a problem with examples
29. Knapsack problem
30. Optimal Binary Search Tree
31. Warshall Algorithm
32. Flyod Algorithm
33. Greedy algorithms
34. Prim’s Algorithm
35. Kruskal’s Algorithm
36. Dijkstra’s Algorithm

# Methods of evaluation

The course is assessed through continuous assessment and a final examination. Continuous assessment will constitute 50% while the final examination will constitute 50% of the final mark. Continuous assessment will be in the form of two group assignments, individual tests, Labs and group project. The final examination will be will be 3 hours.

# Teaching-training activities

-Lectures

- Self-study

- Teacher-led group activities

- Peer assessment

- Presentations