**COURSE CODE: COS431**
**COURSE TITLE:** ALGORITHMS AND COMPLEXITY ANALYSIS
**CREDIT UNITS:** 2


***Course Development Team***

*Dr. Agozie. Eneh* *Course Developer*

*Mrs. Blessing Uzo* *Co-Course Developer*


# COURSE STUDY GUIDE

## 1. Course Information

| | |
|---|---|
| **Course Title**: | Algorithms and Complexity Analysis |
| **Course Code**: | COS431 |
| **Credit Units:** | 3 |
| **Semester Offered:** | First Semester |
| **Course type:** | Major Course |
| **Course Study Time Requirement:** | # Weeks |
| **Number of Modules** | 4 |
| **Number of Study Units** | 12 |

## 2. Course Overview

Computer systems, in terms of hardware and software, have been subjects with the most rapid evolution in the world we live in today. Nevertheless, the demand to deal with complex and challenging requirements of evolving human needs together with the associated human endeavours to meet the needs necessitates the evolution of strategies, solutions, systems, components, and skills that animate these systems, which more often than not are composites of systems with overlapping dependencies. In as much as a system may be viewed from two perspectives; as a set of things working together as parts of a mechanism or an interconnecting network, a complex whole, or as a set of principles or procedures according to which something is done, an organised scheme or method, the fact therefore is that

systems must be given the right level of attention, study, and analysis, which is required for both effective and efficient evolution of solutions and services to meet with the evolving needs of mankind.

As all things stand in the world today, computers are employed in all areas of human endeavour; offices, homes, and schools. Be it in manufacturing, wholesale and retail sales, hospitals, banking, agriculture, security, defence, travel, entertainment, domestic appliances, and even in games. The physical components of the systems (ie hardware) and the systems that operate on them (either as operating systems or application systems) are the objects of everyday livelihood. Historically, the subject of Algorithms, which is a prerequisite to the actual coding of software programs and systems, used to be the concern of computer scientist or mathematicians. However, with the rapid proliferation and ubiquity of computer systems, the subject has become more of a general and highly publicised consideration. It is therefore very important that the subject of algorithms and their analysis, which provides for better solution options in instances of resource limitations, should be part of the fundamental knowledge and skill set of modern day scholars of computing as well as any practitioner of computer science. Therefore, it is somewhat imperative that students and practitioners be able to understand algorithms with respect to how they are created, how they operate, and how they can be critically evaluated in order to be in mainstream computing profession.

## 3. Course Description

After the introduction to algorithms and course overview, the topics to be covered in this course include: Basic Algorithms Analysis; Algorithmic Strategies; Implementation of Data Structures, and Fundamental Computing Algorithms

## 4. Course Aim

This course aims to provide a comprehensive introduction to the study of computer algorithms together with how they can be analysed. Also, the course provides a reasonable number of algorithms and their strategies from which ideas can be generated on how to design new ones or even reuse existing ones.

## 5. Course Objectives

At the end of this course, students should be able to evaluate the role of algorithms in computing, illustrate how to deploy existing algorithms to meet with the requirements of some specified program development, be able to analyse algorithms for correctness and compare similar algorithms for efficiency, and possibly design new or improved algorithms to solve some identified problems.

## 6. Course Prerequisites

The students of this course should have basic programming experience. Students are also required to understand recursive procedures and simple data structures, which are arrays and linked lists. In addition, prior knowledge of basic mathematical proofs, and especially proofs by mathematical induction, is necessary to relate with the areas of verifying algorithms' correctness and those of efficiency considerations and comparisons.

## 7. Course Assessment

| S/N | Assessment Type | Grade % |
|---|---|---|
| | Continuous Assessment | 30% |
| | Examination | 70% |

## 8. Course work Breakdown

| S/N | MODULES | MODULE DESCRIPTION | STUDY UNITS |
|---|---|---|---|
| 1 | Basic Algorithmic Analysis: | Module 1 provides an overview of algorithms, their place in modern computing systems, and how they are analysed. It also defines an algorithm and the asymptotes, and presents some examples together with some justification for | 1. Introduction to Algorithms; Asymptotic Analysis of Upper and Average Complexity Analysis<br><br>2. Standard Complexity Classes; Time and Space Trade-off in Algorithms |

| | | | |
|---|---|---|---|
| | | choosing one algorithm against another and considering them as part of a technology, GUI, OOS, and networks. | Analysis |
| 2. | Algorithmic Strategies | Module 2 delves into some algorithmic techniques. It discusses the concept of recursion as used in divide and conquer algorithms. While the use cases of the various strategies are discussed, the section provides the distinction amongst the strategies and offers ideas about the performance characteristics in terms of their efficiency using the asymptotic notations | 1. Brute Force; Greedy Algorithms<br>2. Divide and Conquer; Backtracking Algorithms<br>3. Dynamic Programming Algorithms<br>4. Heuristics; Reduction (Transform and Conquer) |
| 3 | Implementation of Data Structures | This module discusses the basic sets by which data stored in the computer memory can be manipulated. The sets are held in data structures depending on their intrinsic operability. They can change over time by growing or shrinking. Therefore, this module shows the several types of operations that can be performed on sets, such as insertion, deletion, or test of membership, and | 1. Stacks and Queues<br>2. Linked Lists and Trees |

| | | | |
|---|---|---|---|
| | | illustrates them as they apply to the basic data structures of stacks, queues, linked lists, and trees. | |
| 4 | Fundamental Computing Algorithms | This module introduces the operations often carried out by most applications. They are the sorting and the searching operations. The examples are given together with their use cases with emphasis to the options as regards the growth of functions and sizes of input. Also, the section discusses numerical algorithms that use numerical analysis to solve mathematical problems. Furthermore, the module presents other abstract data structures that are widely used, such as hash tables and graphs, with how they are represented. | 1.Numerical Algorithms<br><br>2. Sequential and Binary Search Algorithms; Binary Search Trees<br><br>3. Sorting Algorithms<br><br>4. Hash Tables; Graphs and its Representation |

**9. References and Further Reading**


i. **Cormen T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., Introduction to Algorithms**, Massachusetts Institute of Technology, Fourth Edition, 2022.

ii. **Sedgewick, R. and Wayne, K., Algorithms**, Addison Wesley, Fourth Edition, 2011.

iii. **Weiss, M. A., Bhattachatjee, A. K., and Mukherjee, S., Data Structures and Algorithms in Java**, Pearson Education Limited, Third Edition, 2012

iv. **Knuth, D. E., The Art of Computer Programming**, Addison Wesley, Volumes 1 to 4, Third Edition, 2011

v. **Heineman, G. T., Pollice G., and Selkow, S., Algorithms in a Nutshell**, O'Reilly Media Inc., Second Edition, 2015

**vi.** And Numerous Internet Resources.

## 10. How to get the Most from this Course

To complete this course, you are required to read the study units and the recommended textbooks and surf the internet for more materials. In this course, each unit may provide self-assessment exercises to test your level of understanding from time to time. At a point in your course, you are required to submit assignments for assessment. At the end of this course there is a final examination. Pay attention to the time recommended for the completion of each study unit and ensure that all tasks and activities are concluded on time. As much as possible, try not to carry over assignments meant to be completed in a week to the next one as this may build up unnecessary pressure on your schedule and could interfere with the learning process.