	THOMAS ADEWUMI UNIVERSITY, OKO-IRESE
Faculty	Computing and Applied Sciences
Department	Mathematical and Computing Science
Program	Computer Science
Course Code	CSC 415
Course Title	**FORMAL MODELS OF COMPUTATION**
Study Year	4
Credit Hours	3
Contact Hours	36
Pre-requisite	
Status	Elective
Semester	First
Mode of Assessment	Lecture, Assessment and Practical
Mode of Delivery	• Classroom Lectures • Laboratory Practical Sessions
Assignment practical **Test** **Examination** **Total**	10% 20% 70% 100%
Course Lecturer and Instructor	
Course Description	Formal models of computation are abstract mathematical frameworks used to study and analyze the behavior of computational systems. These models provide a theoretical foundation for understanding the capabilities and limitations of various computing devices and algorithms.
Course Objectives	To teach the students : • concepts of formal models of computation • the Church-Turing thesis • lambda calculus • Syntax and basic operations • Introduction to finite automata • Deterministic and non-deterministic finite automata • Regular expressions and their equivalence to finite automata
Learning Outcome	At the end of the course, students will be able to: • Different formal models of computation • Use lambda calculus • Describe different types of automata • Introduction to other models of computation (e.g., Petri nets, cellular automata) • Give comparison and analysis of different models • State the applications of alternative models in specific domains
Detailed course contents	Automata theory: Roles of models in computation. Finite-state Automata, Push-down Automata, Formal Grammars, Parsing, Relative powers of formal models.

<table>
<tr><td></td><td colspan="2">Basic_computability: Turing machines, Universal Turing_Machines, Church's thesis, solvability and Decidability.
[Introduction to language structures; languages and their representations; Grammars; Formal notations, types, Chomsky's Language hierarchy; sentence generation and recognition; derivations; Ambiguity and syntax trees; precedence grammars, recognizers. Regular grammar and finite state automata; context-free grammars; Chomsky, Greibach Normal Forms, Push-Down Automata, LR(K), grammars, recursive languages; semantics. Lab. Exercises.]</td></tr>
<tr><td colspan="3">Course Contents Sequencing</td></tr>
<tr><td>Weeks</td><td>Detailed Course Outline</td><td>Allocated Time</td></tr>
<tr><td>WEEK 1, 2</td><td>Introduction to Computability Theory

• Overview of formal models of computation
• Introduction to the Church-Turing thesis
• Turing Machines and their properties
• Decidability and undecidability
• Rice's theorem and the halting problem</td><td>6 Hours</td></tr>
<tr><td>WEEK 3, 4</td><td>Lambda Calculus

• Introduction to lambda calculus
• Syntax and basic operations
• Reduction strategies (e.g., beta-reduction)
• Church encoding and its applications
• Turing-completeness of lambda calculus</td><td>6 Hours</td></tr>
<tr><td>WEEK 5, 6</td><td>Finite Automata and Regular Languages

• Introduction to finite automata
• Deterministic and non-deterministic finite automata
• Regular expressions and their equivalence to finite automata
• Closure properties of regular languages
• Pumping lemma for regular languages

C.A Test</td><td>6 Hours</td></tr>
<tr><td>WEEK 7, 8</td><td>Context-Free Languages and Pushdown Automata

• Introduction to context-free grammars
• Derivations and parse trees
• Pushdown automata and their relation to context-free grammars
• Context-free pumping lemma</td><td>6 Hours</td></tr>
</table>

	• Closure properties of context-free languages	
WEEK 9, 10	Context-Free Languages and Pushdown Automata • Introduction to context-free grammars • Derivations and parse trees • Pushdown automata and their relation to context-free grammars • Context-free pumping lemma • Closure properties of context-free languages	6 Hours
WEEK 11, 12	Other Models of Computation • Introduction to other models of computation (e.g., Petri nets, cellular automata) • Comparison and analysis of different models • Applications of alternative models in specific domains • Limitations and open questions in the field **C.A Test**	6 Hours
	REVISION	

READING LIST:

- Introduction to the Theory of Computation by Michael Sipser
- Computability and Logic by George S. Boolos, John P. Burgess, and Richard C. Jeffrey
- Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman
- Theory of Computation by Dexter C. Kozen
- An Introduction to Formal Languages and Automata by Peter Linz