**Period:** September – December, 2019

|  |  |
| --- | --- |
| **Lecturer:**  **E-mail:**  **Tel:** | Dr. Winfred Yaokumah  wyaokumah@gimpa.edu.gh  024 428 3488 |

**Description**

This course aims to provide a sound basis in the principles of computer science. This includes the study of measuring how long computations may take, probabilistic approaches to difficult problems, the principles of cryptography, the use of grammars to specify syntax rules, formal models of computation, and computability properties. The emphasis is on understanding and application of techniques, rather than formal mathematical proofs. In general, students will learn about a variety of issues in the mathematical development of computer science theory, particularly finite representations for languages and machines, as well as gain a more formal understanding of algorithms and procedures.

**Objectives**

The objective of this course is to develop an understanding of the capabilities and limitations of computer software, based on an understanding of theoretical issues. A number of concrete problems will be addressed in order to develop this understanding at both a conceptual and concrete level. In addressing the problems, students will also gain skills in writing grammars and automata manipulations, regular expressions, proof techniques and reasoning about computational capabilities.

**Learning outcomes**

Upon the completion of this course, the student will be able to:

* Understand the underlying basic concepts of computing and be able to reason about computational capabilities and limitations.
* Be familiar with standard tools and notation for formal reasoning about machines and programs
* Construct finite state machines and the equivalent regular expressions
* Be able to construct pushdown automata and the equivalent context free grammas.
* Be able to prove the equivalence of languages described by pushdown and automata and context free grammas
* Be able to prove the equivalence of languages described by finite state machines and regular expressions

**Broad Topics**

* Theory of Computations
* Normal Forms
* Turing Machines
* Complexity of Computing Problems
* Context-Free Grammars
* Finite State Automata
* Regular Languages and Sets
* Cryptography

**Course Delivery Methods**

The course will be delivered through lectures, practical demonstrations and group discussions.

**Key Texts**

* Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science, Languages and Machines . Addison-Wesley, 3rd edition, 2006. ISBN 0-321-31534-0.
* Gordon, S. (2004) Global Information Technology and E-Business for the Financial Services Industry: Selected Case Studies, Ivy League Publishers, ISBN: 0964838257.

Lewis H. and Papadimitriou C. Elements of the Theory of Computation, Prentice Hall, 2nd edition, 1998. ISBN 0-13-272741-2.

**Measurement of learning outcomes**

* Continuous Assessment (40%)
* End-of-semester examination (60%)

**COURSE CONTENTS AND SCHEDULE**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Theme | Topics | Reading |
|  | Mathematical Preliminaries | 1. *Set Theory* 2. *Cartesian Product, Relations, and Functions* 3. *Equivalence Relations* 4. *Countable and Uncountable Sets* 5. *Directed Graphs Network* 6. *Hardware / software* | Chapter 1: Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science |
|  | Languages | 1. *Strings and Languages* 2. *Finite Specification of Languages* 3. *Regular Sets and Expressions* | CHAPTER 2 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science |
|  | Context-Free Grammars | 1. *Context-Free Grammars and Languages* 2. *Regular Grammars* 3. *Grammars and Languages* 4. *Arithmetic Expressions* | CHAPTER 3 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science |
|  | Normal Forms | 1. *Elimination of Lambda Rules* 2. *Elimination of Chain Rules* 3. *Useless Symbols* 4. *Chomsky Normal Form* 5. *Greibach Normal Form* | *CHAPTER 5 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science* |
| 5. | Finite Automata | 1. *A Finite-State Machine* 2. *Deterministic Finite Automata* 3. *State Diagrams and Examples* 4. *Nondeterministic Finite Automata* 5. *Lambda Transitions* | *CHAPTER 6 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science* |
| 6. | Regular Languages and Sets | 1. *Finite Automata and Regular Sets* 2. *Expression Graphs* 3. *Regular Grammars and Finite Automata* 4. *Closure Properties of Regular Languages* 5. *A Nonregular Language* | *CHAPTER 7 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science* |
| 7. | Pushdown Automata and Context-Free Languages | 1. *Pushdown Automata* 2. *Pushdown Automata and Context-Free Languages* | CHAPTER 8 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science |
| 8. | Turing Machines | 1. *The Standard Turing Machine* 2. *Turing Machines as Language Acceptors* 3. *Multitrack Machines* 4. *Nondeterministic Turing Machines* | *CHAPTER 9 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science* |
| 9. | The Chomsky Hierarchy | 1. *Unrestricted Grammars* 2. *Context-Sensitive Grammars* 3. *The Chomsky Hierarchy* | *CHAPTER 10 Sudkamp, T. Languages and Machines: An Introduction to the Theory of Computer Science* |