Title (Units): **COMP 3180 Theory of Computation (3, 2, 1)**

Course Aims: To introduce students the basic concepts in theoretical computer science, and the formal relationships among machines, languages and grammars.

Prerequisite: MATH 1130 Discrete Structures

Learning Outcomes (LOs):
Upon successful completion of this course, students should be able to:

No.	Learning Outcomes (LOs)
	Knowledge
1	Explain the basic concepts of deterministic and non-deterministic finite automata, regular language, context-free language, Turing machines, Church's thesis, halting problem, computability and complexity
2	Describe the formal relationships among machines, languages and grammars
	Professional Skill
3	Perceive the power and limitation of a computer
4	Solve the problems using formal language
	Attitude
5	Develop a view on the importance of computational theory

Calendar Description: This course aims to introduce the fundamental concepts in theoretical computer science. The topics include deterministic and non-deterministic finite automata, regular language, context-free language, Turing machines, Church's thesis, halting problem, computability, and complexity. Also, the formal relationships between machines, languages and grammars are addressed.

Assessment:

No.	Assessment Methods	Weighting	Remarks
1	Continuous assessment	30%	Continuous assessments are designed to measure how well students have learned the basic concepts and fundamental theories, as well as the use of formal language.
2	Examination	70%	Final examination questions are designed to evaluate how far students have achieved their intended learning outcomes. Students are expected to demonstrate understanding of the theoretical basis of the formal language.

Rubrics: [1/6/2012]

	Excellent (A)	Good (B)	Satisfactory (C)	Marginal Pass (D)	Fail (F)
Formal methods	• Demonstrate a thorough understanding on (i) regular languages and finite state automata, (ii) context free grammars and (iii) Turing machine	• Demonstrate a good understanding on (i) regular languages and finite state automata, (ii) context free grammars and (iii) Turing machine	• Demonstrate a considerable understanding on (i) regular languages and finite state automata, (ii) context free grammars and (iii) Turing machine	• Demonstrate a minimal understanding on (i) regular languages and finite state automata, (ii) context free grammars and (iii) Turing machine	• Unable to demonstrate an understanding on (i) regular languages and finite state automata, (ii) context free grammars and (iii) Turing machine
Computability and complexity theory	• Can describe and explain the concepts of computability • Can analyze the	• Can describe and explain mostly the concepts of computability	• Can describe and explain some concepts of computability • Can analyze the	• Can describe some concepts of computability • Can describe the complexity of a	• Cannot describe the concepts of computability • Cannot describe the complexity

	Excellent (A)	Good (B)	Satisfactory (C)	Marginal Pass (D)	Fail (F)
	complexity of a given problem	• Can analyze the complexity of a given problem with a high degree of effectiveness	complexity of a given problem with some degree of effectiveness	given problem	of a given problem
Problem solving skills	• Can effectively and correctly apply formal methods to solve a given problem	• Can correctly apply formal methods to solve a given problem	• Can apply formal methods to solve a given problem with some degree of effectiveness	• Can apply formal methods to solve a substantial part of a given problem	• Cannot apply formal methods to solve a given problem

Learning Outcomes and Weighting:

Content	**LO No.**
I. Introduction	1
II. Finite Automata and Regular Languages	1-2, 4-5
III. Pushdown Automata and Context-Free Languages	1-2, 4-5
IV. Turing Machines and Phrase-Structure Languages	1-3, 5
V. Computability	1, 3, 5
VI. Complexity	1, 3, 5

References: J. Glenn Brookshear, Theory of Computation: Formal Languages, Automata, and Complexity, Addison-Wesley, 1989.
John C. Martin, Introduction to Languages and Theory of Computation, 3rd Edition, McGraw Hill, 2003.
Peter Linz, An Introduction to Formal Languages and Automata, 4th Edition, 2006.
Hopcroft, Motwani and Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Addison-Wesley, MA 2007.

Course Content in Outline:

Topic

I. Introduction
   A. Review of set theory
   B. Historical background

II. Finite Automata and Regular Languages
   A. Lexical analysis
   B. Deterministic finite automata
   C. Non-deterministic finite automata
   D. Regular grammars and expressions

III. Pushdown Automata and Context-Free Languages
   A. Pushdown automata
   B. Context-free grammars
   C. LL(k) and LR(k) parsers

IV. Turing Machines and Phrase-Structure Languages
   A. Turing machines
   B. Turing-acceptable languages
   C. Beyond phrase-structure languages
   D. Church's thesis
   E. Halting problem

V. Computability
   A. Recursive function theory

B. Primitive recursive functions
C. Partial recursive functions
D. The power of programming languages

VI. Complexity
A. Complexity of computations
B. Complexity of algorithms & problems