==

1. **Name of the course:** Automata Theory and Formal Languages

2. **Objective of the course:** Automata theory deals with the logic of computation with respect to simple machines, referred to as automata. This is an abstract model of machines that perform computations on an input by moving through a series of states or configurations. At each state of the computation, a transition function determines the next configuration on the basis of a finite portion of the present configuration. As a result, once the computation reaches an accepting configuration, it accepts that input. Through this course, the students should be able to understand how machines compute functions and solve problems and more importantly, what it means for a function to be defined as computable or for a question to be described as decidable.

3. Learning Outcome:

The course focuses on three traditionally central areas of the theory of computation: automata, computability and complexity. A student learning this course, should be able to do the following:

(i) Mathematically model machines that perform some dedicated computation (automata theory). These computations may involve some simple operations such as controller of an elevator as well as involved operations like a compiler or hardware design.
(ii) Classify problems as easy ones and hard ones (from the complexity theory),
(iii) Classify problems as solvable or not (in the direction of theories of computability)

4. Detailed Syllabus

Automata and Languages: Finite Automata, Regular Languages, Regular Expressions, Deterministic and Non-deterministic Finite Automata, Minimization of Finite Automata, Closure Properties, Kleene's Theorem, Pumping Lemma and Its Applications, Myhill-Nerode Theorem and Its uses, Decision Algorithms for regular languages; Context-free Grammers, Context-free Languages (CFL), Chomsky Normal Form (CNF), Closure Properties, Pumping Lemma for CFL, Push Down Automata, Acceptance by empty stack and final state, Decision Algorithms for CFL.

Computability: Turing Machine and variants, Computable Functions, Primitive and Recursive Functions, Universality, Halting Problem, Recursive and Recursively Enumerable Sets, Diagonalisation, Reducibility.

Introduction to Complexity: Discussions on Time and Space Complexities, P and NP, NP-completeness, Cook's Theorem, other NP-Complete Problems, PSPACE.

==