RAMAKRISHNA MISSION VIDYAMANDIRA

CBCS Syllabus B.Sc. Computer Science Honours

Semester-V

Course Code: CMSA CC 11 Credit: 6

Course Type: Core Course

Course Outcome:

i) Developing concepts of automaton.

ii) Understanding the various categories of languages and grammars in the Chomsky hierarchy.

iii) Understanding deterministic and nondeterministic finite state automata, and variants of Turing machines.

iv) Developing programming skill in Python.

v) Developing knowledge in working with lists, dictionaries.

vi) Understating use of various scientific libraries in high level programming environment.

CMSA CC 11 T: Theory of Computation

Credit: 4 **Marks: 50**

Finite Automata: Definition of a Finite Automaton, Model, Representation, Classification – with respect to output function Mealy and Moore Machines, with respect to State Transition – Deterministic and Non-Deterministic Machine, Examples, conversion algorithms Mealy to Moore and Moore to Mealy, Finite and Infinite state machines, Finite Automaton, Deterministic and Non-Deterministic Finite automaton, Non-Deterministic to equivalent Deterministic Automaton-Optimized and Non-optimized technique ideas and algorithms, Acceptability of String by a Finite Automaton. [15 L]

Regular Expression: Basic Idea and Definition, Regular Expression basic Identities, Arden's Theorem – Statement (without Proof) and application for reduction of equivalent regular expressions, Regular expression to Finite Automata conversion, State Transition System to Regular Expression conversion algorithm by Arden's Algebraic Method, FA to Regular Grammar and Regular Grammar to FA conversion algorithms and applications, Pumping Lemma. [15 L]

Formal Languages and Grammar: Introduction to Formal Grammar and Language, Chomsky's Classification of Grammar – Type-0, Type-1 or Context Sensitive, Type-2 or Context Free and Type-3 or Regular Grammar, Illustration of each of these classes with example, Sentential form, Sentences – Languages or strings, Derivations. [6 L]

Context Free Language: Context Free Grammar, Parsing, Ambiguity; Normal forms: CNF and GNF, Pumping Lemma.
Definition and basic idea about Push Down Automaton, Definition and basic idea about Linear Bounded Automata. [9 L]

Turing Machine: Concepts of Turing Machine, Formal Definitions, Classifications – Deterministic and Non-Deterministic Turing Machines, Simple Design of Turing Machines: Odd-even count and concepts of Universal Turing Machines, Difference and Similarities between Turing Machine and a General Purpose Computer, Definition and significance of Halting Problem in Turing Machine. [15 L]

CMSA CC 11 P: Python Programming Laboratory

Credit: 2 **Marks: 25**

Introduction to the Python: Variables and the assignment operator, the binding of names to objects and aliasing. Keywords and their significance. [2 L]

Ordered Data types - Strings, Lists and Tuples: Strings: definition, declaration, and immutability, string constants, declaration, and the equivalence of single and double quotes. Multi-line strings. Raw strings. String formatting using the format function and the % operator. f-strings in Python 3.6+. Built-in functions: count, find, replace, upper, lower, strip.
Lists: definition, declaration, and mutability. Nested lists. Indexing and slicing: same as strings.
List comprehensions. The split and join methods. Built-in list functions – append, extend, count, find, index.
Tuples**:** definition, declaration, and immutability; Packing and unpacking lists and tuples.
The + and * operators on strings, lists, and tuples. Indexing and slicing strings, lists, and tuples.
[6 L]

Recapitulation of Programming Fundamentals:
Conditionals and Iterators: Conditionals: If, elif, and else statements. Nested conditionals. Containment checking in containers using the in keyword.
Looping constructs: while and for loops. Flow control using break, continue, and pass. Nested loops.
User-defined Functions and Recursion: Functions: definition, function signature, positional, default, and keyword arguments, Documentation strings.
Recursion: basic idea, implementing recursion, sharing variables across the recursion stack, modifying the size of the recursion stack. [6 L]

File Handling and Exception Handling: open and close methods, the different read and write modes. Using the with open approach to files. read, readline, readlines functions.
Exception handling: the popular errors- NameError, ValueError, SyntaxError, KeyError, AttributeError and their cause and effects. Using try-except blocks for graceful handling of exceptions. [5 L]

Unordered data types - Sets and Dictionaries: Basic concepts of hashing: hash functions, open chain, closed chain, advantages and disadvantages compared to conventional ordered data types. The hash() function in Python.
Sets and frozensets: definition, declaration, mutability, and advantages over lists / tuples.
Insertion, deletion, union, intersection, and other built-in operations. [5 L]

Dictionaries: Concept of keys and values. Immutability requirement for keys. Basic operations on dictionaries. Iterating over the keys and key, value pairs of a dictionary. Dictionary inversions.

Installation and basics of various Python packages like numpy, scipy, networkx, matplotlib. [6 L]

Introduction to Python based Web development : Django framework [10 L]

Recommended Books:

1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 3rd Edition, Pearson.
2. Introduction to Theory of Computation by Micheal Sipser, 3rd Edition, Cengage Learning.
3. Theory of Computer Science (Automata, Languages & Computation) by K L P Misra & N Chandrasekharan, 3rd Edition, PHI.
4. An Introduction to Formal Languages and Automata by Peter Linz, 4th Edition, Narosa.
5. Introduction to Computation and Programming Using Python: With Application to Understanding Data by John V Guttag, 2nd Edition, MIT.
6. Learn Python3 the Hard Way by Zed Shaw, Addison Wesley.