Introduction to Computing

DCA-102

BLOCK I

Unit 1: Computing Computing: Processes, Procedures, and Computers; Measuring Computing Power: Information, Representing Data, Growth of Computing Power; Science, Engineering, and the Liberal Arts, Summary and Roadmap.

Unit II: Defining Procedures Language, Surface Forms and Meanings, Language Construction , Recursive Transition Networks, Replacement Grammars.

Unit III: Programming Problems with Natural Languages, Programming Languages, Scheme, Expressions: Primitives, Application Expressions; Definitions, Procedures: Making Procedures, Substitution Model of Evaluation; Decisions, Evaluation Rules.

BLOCK II

Unit IV: Problems and Procedures Solving Problems, Composing Procedures: Procedures as Inputs and Outputs; Recursive Problem Solving, Evaluating Recursive Applications, Developing Complex Programs: Printing, Tracing .

Unit V: Data Data Types, Pairs: Making Pairs, Triples to Octuples; Lists, List Procedures: Procedures that Examine Lists, Generic Accumulators, Procedures that Construct Lists; Lists of Lists, Data Abstraction.

Unit VI: Analyzing Procedures Machines, History of Computing Machines, Mechanizing Logic: Implementing Logic, Composing Operations, Arithmetic; Modeling Computing: Turing Machines .

BLOCK III

Unit VII: Cost Empirical Measurements, Orders of Growth: Big O, Omega Theta; Analyzing Procedures: Input Size, Running Time, Worst Case Input; Growth Rates: No Growth: Constant Time, Linear Growth, Quadratic Growth, Exponential Growth, Faster than Exponential Growth , Non- terminating Procedures

Unit VIII: Sorting and Searching Sorting: Best-First Sort, Insertion Sort , Quicker Sorting, Binary Trees, Quicksort; Searching: Unstructured Search, Binary Search, Indexed Search;

Unit IX: Improving Expressiveness Mutation, Assignment, Impact of Mutation: Names, Places, Frames, and Environments, Evaluation Rules with State; Mutable Pairs and Lists, Imperative Programming: List Mutators, Imperative Control Structures

BLOCK IV

Unit X: Objects Packaging Procedures and State: Encapsulation, Messages, Object Terminology; Inheritance:

Implementing Subclasses, Overriding Methods; Object-Oriented Programming.

Unit XI: Interpreters Python: Python Programs, Data Types, Applications and Invocations, Control Statements; Parser, Evaluator: Primitives, If Expressions, Definitions and Names, Procedures , Application , Finishing the Interpreterl Lazy Evaluation: Lazy Interpreter, Lazy Programming.

Unit XII: The Limits of Computing Computability, Mechanizing Reasoning: Godel's Incompleteness Theorem; The Halting Problem, Universality, Proving Non-Computability.

Supplementary Course Material available at: http://www.computingbook.org/FullText.pdf