



# PROGRAMMING, DATA STRUCTURES AND ALGORITHMS USING PYTHON

## PROF. MADHAVAN MUKUND

Department of Computer Science and Engineering  
Chennai Mathematical Institute

**INTENDED AUDIENCE :** Students in any branch of mathematics/science/engineering, 1st year

**PRE-REQUISITES :** School level mathematics.

**INDUSTRIES APPLICABLE TO :** This course should be of value to any company requiring programming skills.

## COURSE OUTLINE :

This course is an introduction to programming and problem solving in Python. It does not assume any prior knowledge of programming. Using some motivating examples, the course quickly builds up basic concepts such as conditionals, loops, functions, lists, strings and tuples. It goes on to cover searching and sorting algorithms, dynamic programming and backtracking, as well as topics such as exception handling and using files. As far as data structures are concerned, the course covers Python dictionaries as well as classes and objects for defining user defined datatypes such as linked lists and binary search trees.

## ABOUT INSTRUCTOR :

Prof. Madhavan Mukund studied at IIT Bombay (B.Tech) and Aarhus University (PhD). He has been a faculty member at Chennai Mathematical Institute since 1992, where he is presently Professor and Dean of Studies. His main research area is formal verification. In addition to the NPTEL MOOC programme, he has been involved in organizing IARCS Instructional Courses for college teachers. He is a member of ACM India's Education Committee. He has contributed lectures on algorithms to the Massively Empowered Classroom (MEC) project of Microsoft Research and the QEEE programme of MHRD.

## COURSE PLAN :

**Week 01 :** Informal introduction to programming, algorithms and data structures via gcd, Downloading and installing Python, gcd in Python: variables, operations, control flow - assignments, conditionals, loops, functions.

**Week 02 :** Python: types, expressions, strings, lists, tuples | Python memory model: names, mutable and immutable values | List operations: slices etc| Binary search | Inductive function definitions: numerical and structural induction | Elementary inductive sorting: selection and insertion sort | In-place sorting.

**Week 03 :** Basic algorithmic analysis: input size, asymptotic, complexity,O() notation | Arrays vs lists | Merge sort | Quicksort | Stable sorting.

**Week 04 :** Dictionaries | More on Python functions: optional arguments, default values | Passing functions as arguments | Higher order functions on lists: map, iter, list comprehension.

**Week 05 :** Exception handling | Basic input/output | Handling files | String processing.

**Week 06 :** Backtracking: N Queens, recording all solutions | Scope in Python: local, global, nonlocal names | Nested functions | Data structures: stack, queue | Heaps.

**Week 07 :** Abstract datatypes | Classes and objects in Python | "Linked" lists: find, insert, delete | Binary search trees: find, insert, delete | Height-balanced binary search trees.

**Week 08 :** Efficient evaluation of recursive definitions: memoization | Dynamic programming: examples | Other programming languages: C and manual memory management | Other programming paradigms: functional programming.