# Six Phrase – MySlate – Technical Training Syllabus

## Table of Contents

SEMESTER 1 – Problem Solving Skills Using C Programming .................... 2

SEMESTER 2 - Advanced C Programming ...................................................... 4

SEMESTER 3 – Problem Solving using Basic Data Structures .................... 6

SEMESTER 4 – Basic Design and Analysis of Algorithms............................. 8

SEMESTER 5 – Advanced Data Structures .................................................... 10

SEMESTER 6 – Advanced Design and Analysis of Algorithms .................. 12

OPTIONAL COURSE – Full Stack Development using Java/J2EE – Part 1 ........................................................................................................................ 14

OPTIONAL COURSE – Full Stack Development using Java/J2EE – Part 2 ........................................................................................................................ 16

OPTIONAL COURSE – Data Science & Artificial Intelligence using Python – Part 1 ....................................................................................... 18

OPTIONAL COURSE – Data Science & Artificial Intelligence using Python – Part 2 ....................................................................................... 20

# SEMESTER 1 – Problem Solving Using C Programming

**COURSE OBJECTIVES**

The course aims to provide exposure to problem-solving through programming. It aims to train the student to the basic concepts of the C-programming language. This course involves a lab component which is designed to give the student hands-on experience with the concepts.

**COURSE OUTCOMES**

**Upon completion of this course, the students will be able to:**

- Identify situations where computational methods and computers would be useful.
- Given a computational problem, identify and abstract the programming task involved.
- Approach the programming tasks using techniques learned and write pseudo-code.
- Choose the right data representation formats based on the requirements of the problem.
- Use the comparisons and limitations of the various programming constructs and choose the right one for the task in hand.
- Write the program on a computer, edit, compile, debug, correct, recompile and run it.
- Identify tasks in which the numerical techniques learned are applicable and apply them to write programs, and hence use computers effectively to solve the task.

## UNIT 1: INTRODUCTION TO PRINCIPLES OF PROGRAMMING:

Introduction to Programming , Programing Domain : Scientific Application , Business Applications, Artificial Intelligence, Systems Programming , Web Software Categories of Programming Languages: Machine Level Languages, Assembly Level Languages , High Level Languages Programming Design Methodologies : Top Down and Bottom UP Program Development Cycle with case study, Program Execution and Translation Process ,Problem solving using Algorithms and Flowcharts, Performance Analysis and Measurements: Time and Space complexity.

## UNIT 2: INTRODUCTION TO C PROGRAMMING:

Features of C and its Basic Structure, Simple C programs, Constants, Integer Constants, Real Constants, Character Constants, String Constants, Backslash Character Constants, Concept of an Integer and Variable, Rules for naming Variables and assigning values to variables, Floating-point Numbers, Converting Integers to Floating-point and vice-versa, Mixed-mode Expressions, The type cast Operator, The type char, Keywords, Character Input and Output, Formatted input and output, The gets() and puts() functions, Interactive Programming.

## UNIT 3: OPERATORS, EXPRESSIONS AND CONTROL STATEMENTS:

Arithmetic Operators, Unary Operators, Relational and Logical Operators, The Conditional Operator, Library Functions, Bitwise Operators, The Increment and Decrement Operators, The Size of Operator, Precedence of operators, The goto statement, The if statement, The if-else statement, Nesting of if statements, The conditional expression, The switch statement, The while loop, The do…while loop, The for loop, The nesting of for loops, The break statement and continue statement.

## UNIT 4: ARRAYS, STRINGS AND POINTERS:

One Dimensional Arrays, Passing Arrays to Functions, Multidimensional Arrays, Strings, Basics of Pointers, Pointers and One-dimensional Arrays, Pointer Arithmetic, Pointer Subtraction and Comparison, Similarities between Pointers and One-dimensional Arrays, Null pointers, Pointers and Strings, Pointers and two-dimensional arrays, Arrays of Pointers.

## UNIT 5: STRUCTURES, UNIONS AND FUNCTIONS

Basics of Structures, Arrays of Structures, Pointers to Structures, Self-referential Structures, Unions, Function Philosophy, Function Basics, Function Prototypes, and Passing Parameters: Passing Parameter by value and Passing Parameter by reference, passing string to function, Passing array to function, Structures and Functions Recursion.

## REFERENCES:

1. Programming in ANSI C - Balagurusamy - Tata McGraw-Hill Education, 2008
2. Programming in C (3rd Edition), by Stephen G. Kochan, Sams, 2004
3. Programming in C - Stephen G. Kochan, III Edition, Pearson Education.

# SEMESTER 2 - Advanced C Programming

## COURSE OBJECTIVES

The course is oriented to those who want to advance structured and procedural programming understating and to improve C programming skills. The major objective is to provide students with understanding of code organization and functional hierarchical decomposition with using complex data types..

## COURSE OUTCOMES

**Upon completion of this course, the students will be able to:**

- Understanding a functional hierarchical code organization.
- Ability to define and manage data structures based on problem subject domain.
- Ability to work with textual information, characters and strings. • Ability to work with arrays of complex objects.
- Understanding a concept of object thinking within the framework of functional model.
- Understanding a concept of functional hierarchical code organization.
- Understanding a defensive programming concept. Ability to handle possible errors during program execution.

## UNIT 1: INTRODUCTION TO RECURSION:

Introduction to Recursion, Types of Recursion - Head Recursion , Tail Recursion, Tree Recursion, Indirect Recursion and Nested Recursion . Recursion vs Looping - Analysis on efficiency of looping and recursion, Working of recursive code in main memory. Recurrence Relation , Different types of recurrence relation. Deriving time complexity and space complexity using recurrence relation.

## UNIT 2: GROWTH FUNCTIONS AND RECURSION:

Polynomial Equations, Compare growth functions - order growth functions, omega growth functions, theta growth functions - Constant time, Linear time, Logarithmic time, Quadratic time and exponential time . Problems on Recursions - Factorial Number, Sum of first N Natural Numbers, Nth Fibinocci Number, Exponent Function, Taylor Series, Tower of Hanoi.

## UNIT 3: STORAGE CLASSES, THE PREPROCESSOR AND DYNAMIC MEMORY ALLOCATION:

Storage Classes and Visibility, Automatic or local variables, Global variables, Static variables, External variables, File Inclusion, Macro Definition and

Substitution, Macros with Arguments, Nesting of Macros, Conditional Compilation, Dynamic Memory Allocation, Allocating Memory with malloc, Allocating Memory with calloc, Freeing Memory, Reallocating Memory Blocks, Pointer Safety, The Concept of linked list, Inserting a node by using Recursive Programs, Sorting and Reversing a Linked List, Deleting the Specified Node in a Singly Linked List.

## UNIT 4: FILE MANAGEMENT:

Defining and Opening a file, Closing Files, Input/output Operations on Files, Predefined Streams, Error Handling during I/O Operations, Random Access to Files, Command Line Arguments.

## UNIT 5: BIT MANIPULATION

The hexadecimal number system, C bitwise operators, Working with individual bits, How to check if a given number is a power of 2, Count the number of ones in the binary representation of the given number, Check if the $i^{th}$ bit is set in the binary form of the given number, How to generate all the possible subsets of a set, Find the largest power of 2 (most significant bit in binary form), which is less than or equal to the given number N, Tricks with Bits, Applications of bit operations.

**REFERENCES:**

1. R. G. Dromey, "How to Solve It By Computer", Pearson, 1982
2. A.R. Bradley, "Programming for Engineers", Springer, 2011
3. Kernighan and Ritchie, "The C Programming Language", (2nd ed.) Prentice Hall, 1988

# SEMESTER 3 – Problem Solving using Basic Data Structures

**COURSE OBJECTIVES**

The objective of the course is to familiarize students with basic data structures and their use in fundamental algorithms.

**COURSE OUTCOMES**

**Upon completion of this course, the students will be able to:**

- Data abstraction and information hiding.
- linear data structures and their applications in problem solving and programming.
- Nonlinear data structures and their applications in problem solving and programming.
- Internal and external sort and search techniques.

## UNIT 1: LINKED LIST & STACK:

Linked List - Singly Linked List, Structure, Node creation, Singly Linked List Representation, Singly Linked List Traversal, Doubly Linked List - Structure, Node creation, Doubly Linked List Representation, Doubly Linked List Traversal, Circular Linked List, Structure, Node creation, Circular Linked List Representation, Circular Linked List Traversal Stack, Stack – Introduction Push, Pop, Peek or Top, isEmpty() and isFull(), Time Complexities of the Operations, Applications of Stack Balancing of symbols, Infix to Postfix/Prefix Conversation, Stock Span Problem, Histogram Problem, Implementation - Using Array, Using Linked List

## UNIT 2: QUEUE & HEAP:

Queue - Introduction - EnQueue, Applications of Queue Data Structure, Priority Queue, Applications of Priority Queue, Deque, Circular Queue, Implementation - Queue using Stack, LRU cache Implementation, Stack using queue, Heap - Binary Heap, Complete Binary Tree, Tree Representation of Binary Heap, Max Binary Heap, Min Binary Heap, Insertion and Deletion in Binary Heap, Heap Sort, Application of Binary Heap.

## UNIT 3: BINARY TREE:

Binary Tree - Introduction, What is Tree?, Representation of Tree, Properties and Terminologies of Binary Tree, Types of Binary Tree Full ,Complete,Perfect and Balanced Tree, Degenerate or pathological Tree, Binary Search Tree, Insertion and Deletion, Inorder , Preorder , PostOrder and LevelOrder Traversal

## UNIT 4: TREES AND HASHING:

Hashing - Hashing Techniques, Hashing Introduction, Linear Probing for Collision Handling, Separate Chaining for Collision Handling, Open Addressing for Collision Handling, Union and Intersection of two Linked Lists. AVL Tree Left-Left, Left-Right, Right-Right and Right-Left Imbalance, Left and Right Rotation, AVL with duplicate keys, Implementation - Red Black Tree, Rules of coloring in Red-black Tree, Left and Right Rotation, Implementation of Insertion and Deletion

## UNIT 5: GRAPH

Graph terminology –Representation of graphs –Path matrix –Graph Traversal –BFS (breadth first search) –DFS (depth first search) –Minimum spanning Tree –Kruskal's Algorithm & Prim's Algorithm –Warshall's algorithm (shortest path algorithm).

### REFERENCES:

1. Weiss, Mark. A. (2012), *Data structures and algorithm analysis in Java*. 3 edition. Harlow, Essex : Pearson (632 p).
2. Zobel, Justin (2014), *Writing for Computer Science*. 3 edition. Springer Verlag London Ltd (270 p).

# SEMESTER 4 – Basic Design and Analysis of Algorithms

## COURSE OBJECTIVES

This course assumes that students know how to analyze simple algorithms and data structures. It introduces students to the design of computer algorithms, as well as analysis of sophisticated algorithms.

## COURSE OUTCOMES

**Upon completion of this course, the students will be able to:**

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

## UNIT 1: INTRODUCTION TO ALGORITHMS:

Analysis of Algorithms - Asymptotic Analysis, Worst, Average and Best Cases, Asymptotic Notation, little o, and little omega notations, Lower and Upper Bound Theory, Analysis of Loops, Solving Recurrences, Amortized Analysis, What does 'Space Complexity' mean?, Pseudo-polynomial Algorithms, NP-Completeness Introduction, Polynomial Time Approximation Scheme, A Time Complexity Question, Time Complexity of building a heap, Time Complexity where loop variable is incremented by 1, 2, 3, 4 .., Time Complexity of Loop with Powers, Performance of loops (A caching question).

## UNIT 2: SEARCHING AND SORTING:

Searching - Linear Search, Binary Search, Jump Search, Interpolation Search, Exponential Search, Ternary Search, Sorting - Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Heap Sort, QuickSort, Radix Sort, Counting Sort, Bucket Sort, ShellSort, Comb Sort, Pigeonhole Sort, Cycle Sort, Interpolation search vs Binary search, Stability in sorting algorithms, Lower bound for comparison-based sorting algorithms, Merge Sort for Linked Lists, Iterative Quick Sort, QuickSort on Singly Linked List, QuickSort on Doubly Linked List, A Problem in Many Binary Search Implementations, Sort an array in waveform, Why is Binary Search preferred over Ternary Search?

## UNIT 3: BACKTRACKING

Print all permutations of a given string and number, Knight's Tour Plan, Rat in a Maze, N Queen Problem, Subset Sum, m-Coloring Problem, Hamiltonian Cycle, Sudoku, Tug of War.

## UNIT 4: BRANCH AND BOUND:

Introduction with 0/1 Knapsack, Implementation of 0/1 Knapsack, 8 puzzle Problem, Job Assignment Problem, N Queen Problem, Traveling Salesman Problem.

## UNIT 5: MISC, DIVIDE AND CONQUER:

Misc -Sieve of Eratosthenes, Sieve of Sundaram, Fermat Little Theorem, Even Fibonacci Series.

Divide and Conquer - Median of two sorted arrays, Count Inversions, Closest Pair of Points, Strassen's Matrix Multiplication.

### REFERENCES:

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. "Algorithms Unlocked" by Thomas H. Cormen.

# SEMESTER 5 – Advanced Data Structures

**COURSE OBJECTIVES**

The objective of the course is to familiarize students with basic data structures and their use in fundamental algorithms.

**COURSE OUTCOMES**

**Upon completion of this course, the students will be able to:**

- Data abstraction and information hiding.
- linear data structures and their applications in problem solving and programming.
- Nonlinear data structures and their applications in problem solving and programming.
- Internal and external sort and search techniques.

## UNIT 1: SEGMENT TREE:

Segment Tree - Sum of given Range, Representation of Segment Tree for Sum Range Query, Implementation of Sum Range Query, Minimum Range Query, Representation of Segment Tree for Minimum Range Query, Implementation of Minimum Range Query. Lazy Propagation in Segment Tree - Introduction, Implementation,. Persistent Segment Tree - Introduction, Implementation.

## UNIT 2: TRIE, SUFFIX ARRAY AND SUFFIX TREE:

Trie - Introduction, Representation of Trie, Operations - Insert , Search and Delete, Implementation, Longest Prefix Matching, Print unique rows in a given boolean matrix, Suffix Array and Suffix Tree, Suffix Array - Introduction, nLogn Algorithm, Kasai's Algorithm for Construction of LCP array from Suffix Array, Suffix Tree - Introduction , Ukkonen's Suffix Tree Construction, Generalized Suffix Tree, Creating a Linear time Suffix Array using Suffix Tree, Substring Check, Searching All Patterns, Longest Repeated Substring, Longest Common Substring

## UNIT 3: SPLAY TREE AND B - TREE:

Splay Tree - Introduction, Representation of Splay Tree, Operations - Insert , Search and Delete, Implementation - B Tree, Introduction, Disk Structure, Level Indexing, Multi-Level Indexing, Implementation, Insertion and Deletion in B Tree.

## UNIT 4: GRAPH I:

Graph - Introduction, Graph and its Representations, Breadth First

Traversal for a Graph, Depth First Traversal for a Graph, Implementation - Topological Sorting, BiPartite Graph, Snake and Ladder Problem, Spanning Tree - Minimum Spanning Tree using Kruskal Algorithm, Minimum Spanning Tree using Prim's Algorithm, Boruvka's Algorithm for Minimum Spanning Tree, Steiner Tree

## UNIT 5: GRAPH II

Graph Cycle - Detect cycle in Directed and Undirected graph, Longest Path in a Directed Acyclic Graph, Disjoint set or Union Find, Union Find Algorithm ( Union By Rank and Find by Optimized Path Compression ), Shortest Paths - Dijkstra's shortest path Algorithm, Bellman-Ford Algorithm, Floyd Warshall Algorithm, Johnson's Algorithm for All-pairs shortest paths, Maximum Flow - Ford-Fulkerson Algorithm for Maximum Flow Problem, Find minimum s-t cut in a flow network, Maximum Bipartite Matching, Karger's Algorithm, Dinic's Algorithm.

**REFERENCES:**

1. Weiss, Mark. A. (2012), *Data structures and algorithm analysis in Java*. 3 edition. Harlow, Essex : Pearson (632 p).
2. Zobel, Justin (2014), *Writing for Computer Science*. 3 edition. Springer Verlag London Ltd (270 p).

# SEMESTER 6 – Advanced Design and Analysis of Algorithms

## COURSE OBJECTIVES

This course assumes that students know how to analyze simple algorithms and data structures. It introduces students to the design of computer algorithms, as well as analysis of sophisticated algorithms.

## COURSE OUTCOMES

**Upon completion of this course, the students will be able to:**

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

## UNIT 1: GREEDY ALGORITHMS:

Activity Selection Problem, Kruskal's Minimum Spanning Tree Algorithm, Huffman Coding, Prim's Minimum Spanning Tree, Dijkstra's shortest path Algorithm, Job Sequencing Problem, Coin change Problem, K centers Problem, Minimum Number of Platforms Required for a Railway/Bus Station

## UNIT 2: DYNAMIC PROGRAMMING 1:

Overlapping Subproblems Property, Optimal Substructure Property, Longest Increasing Subsequence, Longest Common Subsequence, Edit Distance, Min Cost Path, Coin change Problem, Matrix Chain Multiplication, Binomial Coefficient, 0-1 Knapsack Problem, Egg Dropping puzzle.

## UNIT 3: DYNAMIC PROGRAMMING 2:

Longest Palindromic Subsequence, Cutting a Rod, Maximum Sum Increasing Subsequence, Longest Bitonic Subsequence, Floyd Warshall Algorithm, Palindrome Partitioning, Partition Problem, Word Wrap Problem, Box Stacking, Maximum Size square Sub-Matrix with all 1s, Ugly Numbers, Largest Sum Contiguous SubArray, Longest Palindromic Substring, Bellman-Ford Algorithm, Largest Independent Set Problem, Subset Sum, Maximum Sum Rectangle in a 2D Matrix.

## UNIT 4: PATTERN SEARCHING:

Naïve Pattern Searching, KMP Algorithm, Rabin-Karp Algorithm, Boyer Moore Algorithm, Anagram Substring Search, Aho-Corasick Algorithm, Kasai's Algorithm for Construction of LCP array from Suffix Array, Z algorithm, Manacher's Algorithm

## UNIT 5: ANALYSIS OF ALGORITHMS:

Comparison of time and space complexities of all sorting and searching algorithms, KMP vs Rabin-Karp vs Boyer Moore, Solving problems in Greedy vs Dynamic Programming, Quick Sort vs Merge Sort,

**REFERENCES:**

1. “Introduction to Algorithms” by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. “Algorithms Unlocked” by Thomas H. Cormen.

# Semester 7 - ELECTIVE COURSE – Full Stack Development using Java/J2EE – Part 1

## COURSE OBJECTIVES

Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc. Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc and exception handling mechanisms. Understand the principles of inheritance, packages and interfaces

## COURSE OUTCOMES

**Upon completion of this course, the students will be able to:**

- Identify classes, objects, members of a class and relationships among them needed for a specific problem
- Write Java application programs using OOP principles and proper program structuring
- Demonstrate the concepts of polymorphism and inheritance.
- Write Java programs to implement error handling techniques using exception handling

## UNIT 1: INTRODUCTION, DATA TYPES AND STRINGS:

Data Types - Boolean , Byte , Character , Short , Integer , Long , Float and Double Operators - Unary , Arithmetic , Shift , Relational , Bitwise , Logical and Ternary Strings Concatenation and Comparison - charAt() , compareTo() , concat() , contains() , endsWith() and equals(), IgnoreCase() , format() , getBytes() , getChars() , indexOf() and intern(), isEmpty() , join() , lastIndexOf() , length() , replace() and replaceAll(), split() , startsWith() , substring() and toCharArray(), toLowerCase() , toUpperCase() , trim() and valueOf().

## UNIT 2: DECISION MAKING , LOOPING & Arrays:

Decision Making - if, else-if, nested-if, else, if-elseif, break, continue, return Looping - switch, while, do-while, for, for-each. Collections:List - AbstractList, AbstractSequentialList, ArrayList, LinkedList, Immutable List, Collections:Queue AbstractQueue - ArrayBlockingQueue, ConcurrentLinkedQueue, LinkedBlockingQueue, LinkedTransferQueue, PriorityBlockingQueue, Collections:Deque -ArrayDeque, ConcurrentLinkedDeque, LinkedBlockingDeque.

## UNIT 3: COLLECTIONS:

Collections:Set - AbstractSet, EnumSet, HashSet, TreeSet, LinkedHashSet, Collections:Map - EnumMap, HashMap, SortedMap, WeakHashMap, ImmutableMap, LinkedHashMap, IdentityHashMap, Dictionary, Collections:HashTable - Introduction, Collections:Stack - Introduction, Collections:Vector - Introduction.

## UNIT 4: OBJECT ORIENTED PROGRAMMING:

Object Oriented Programming - object and class, Instance attribute, class attribute or static attribute, Instance method, class method or static method, constructors, self, Inheritence, Types of Inheritence, Single Inheritence, Multilevel Inheritence, Multiple Inheritence, Hybrid Inheritence, Data encapsulation and Abstraction, Polymorphism, Method OverRiding, Operator overloading, Operator overRiding, init constructor, str constructor, repr constructor, Interface, Abstract class, Inner class, Nested class.

## UNIT 5: DATASTRUCTURE AND ALGORITHM WITH JAVA

Datastructure - stack, queue, Singly ,Doubly and Circular Linked List, Binary , Binomial and Fibonacci Heap, Binary Search Tree, AVL Tree, RedBlack Tree, B Tree, Splay Tree, Segment Tree. Algorithms - Searching and Sorting, Pattern Matching, BackTracking, Greedy, Dynamic Programming, Divide and Conquer.

**REFERENCES:**

1. E Balagurusamy, “Programming with Java A Primer”, TMH, 4th edition.
2. Patrick Naughton, “Java Handbook”, Osborne McGraw-Hill

# Semester 7 - ELECTIVE COURSE – Full Stack Development using Java/J2EE – Part 2

## COURSE OBJECTIVES

The objective of this course is to provide the necessary knowledge to design and develop dynamic, database-driven application using J2EE. Students will learn how to connect to any JDBC-compliant database, and perform hands-on practice with a database to create database-driven connectivity.

## COURSE OUTCOMES

**Upon completion of this course, the students will be able to:**

- Knowledge of the structure and model of the Java programming language.
- Apply the concepts of JDBC, Transaction processing, statement objects and Resultset to perform operations on Database
- Develop distributed web applications using RMI, Servlets and JSP.
- Use the Java programming language for various programming technologies.
- Apply the concept of Cookies, Session, Enterprise JavaBeans and JAR files to develop server-side applications
- Choose an engineering approach to solving problems, starting from the acquired knowledge of programming.

## UNIT 1: JAVA DATABASE CONNECTIVITY:

JDBC : Introduction, JDBC Drivers,DB Connectivity Steps, Connectivity with Oracle, DriverManager, ConnectionStatement, ResultSet, Prepared Statement, ResultSetMetaData, CallableStatement, Batch Processing.

## UNIT 2: SERVLETS AND JSP:

Introduction to Java Web Application, Servlet : Session Management, Servlet Filter, Servlet Listener, Cookies , Exception Handling, Servlet Database Connection, Tomcat DataSource JNDI Example. JSP : Implicit Objects, Directives, Exception Handling, Expression Language, Action Tags, JSTL, JSP Custom Tags.

## UNIT 3: STRUTS2:

Introduction, Struts2 Features, Model1 vs Model2. Core Components : Interceptors, ValueStack, ActionContext, ActionInvocation. Architecture, Action, Configuration. Interceptors : Custom Interceptor, params interceptor, execAndWait, prepare interceptor, modelDriven interceptor, exception interceptor, fileUpload interceptor, Struts2 Validation.

## UNIT 4: HIBERNATE:

Introduction, Architecture, Hibernate Example : Web application, Generator classes, Dialects. Hibernate Log4j : HB with Log4j 1, HB with Log4j 2.
Inheritance Mapping: Inheritance Mapping, Table Per Hierarchy, TPH using Annotation, Table Per Concrete, TPC using Annotation, Table Per Subclass, TPS using Annotation. Hibernate Mapping: Collection Mapping, Mapping List, Mapping Bag, Mapping Set, Mapping Map, One To Many XML, One To Many Annotation, Many To Many XML, Many To Many Annotation. Tx Management, HQL, HCQL, Named Query, Hibernate Caching, Second Level Cache.

## UNIT 5: APACHE TOMCAT, WEBSERVICE AND JAVABEAN:

Introduction, Installation, Managing Apache Tomcat: Starting, Testing, Admin Console, Development.Tomcat as Server. JavaBean: Session Bean
Session Bean, Stateless Session Bean, Stateful Session Bean. JMS : JMS, JMS Queue, JMS Topic. Message Driven Bean, Entity Bean.
Web Services : WS Components, SOAP Web Service, RESTful Web Service, SOAP vs RESTSOA.

**REFERENCES:**

1. James Edward Keogh, “J2EE: The complete Reference”.
2. Kathy Sierra, Bert Bates, “SCJP Sun Certified Programmer for Java 6”.

# Semester 8 - ELECTIVE COURSE – Data Science & Artificial Intelligence using Python – Part 1

**COURSE OBJECTIVES**

The course is designed to provide Strong knowledge of Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.

**COURSE OUTCOMES**

**Upon completion of this course, the students will be able to:**

- Explain basic principles of Python programming language
- Implement object oriented concepts,
- Implement database and GUI applications.

## UNIT 1: INTRODUCTION, DATATYPES AND STRINGS:

DataTypes - Integer , Float , Boolean , String , List , Tuple , Dictionary and Sets. String - Concatenation and Replication, find() , split() and replace(), upper() ,isupper() , lower() and islower() functions, swapcase() , capitalize() and title() functions, isalpha() ,isdigit() / isnumeric() ,isalnum functions, Slicing Operation

## UNIT 2: LIST & TUPLES:

List - Concatenation , Replication , Slicing Operation, sort() , sorted() , reverse() and reversed() functions, count() , index() , insert() , remove() and append() functions, pop() , del() , max() , min() and join() functions, Shallow Copy and Deep Copy, Packing and unpacking of data in a list, Nested List, Tuple - Concatenation , Replication , Slicing Operations, sorted() , reversed() , min() , max() , index() and count() function, packing and unpacking of data in a tuple

## UNIT 3: DICTIONARY AND SETS:

Dictionary - Creating a Dictionary, Nesting a Dictionary, Inserting a key in a Dictionary, Accessing and Deleting elements in a Dictionary, del Keyword, pop() , popitem() , clear() , copy() and get() methods, dictionary_name.values(), update() , setdefault() , keys() , items() , has_key() and fromkeys(), type() and cmp(). Sets - Frozen sets, Internal working of sets, add() , union() , intersection() and difference() method, symmetric_difference, clear() method, Operators in sets

## UNIT 4: FUNCTIONS, LIBRARY AND EXCEPTIONAL HANDLING:

Higher Order Functions - map , filter , reduce and lambda function, Random Library - randint() , randrange() , choice() and shuffle() functions, random() and uniform() functions, Math Library - sqrt() , pow() , gcd() ,factorial and fabs() functions, floor() , ceil() , log() , log2() and log10() functions, Collections Module namedtuple() , deque() , Counter() , OrderedDict and defaultdict, Exception Handling - All Error Categories, try , except , finally blocks, Raising an exception

## UNIT 5: REGULAR EXPRESSIONS AND OBJECT ORIENTED PROGRAMMING

Regular Expression - compile() , split() , sub() , subn() and escape() functions, meta characters.

Object Oriented Programming - object and class, Instance attribute, class attribute or static attribute, Instance method, class method or static method, constructors, self, Inheritence, Types of Inheritence, Single Inheritence, Multi-Level Inheritence, Multiple Inheritence, Hybrid Inheritence, Data encapsulation and Abstraction, Polymorphism, Method OverRiding, Operator overloading, operator overRiding, init constructor, str constructor, repr constructor

**REFERENCES:**

1. Sebastian Raschka, and Vahid Mirjalili, Python Machine Learning
2. Alberto Artasanchez, Prateek Joshi (2020), Artificial Intelligence with Python — Second Edition

# Semester 8 - ELECTIVE – Data Science & Artificial Intelligence using Python – Part 2

**COURSE OBJECTIVES**

The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches. Develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning.

**COURSE OUTCOMES**

**Upon completion of this course, the students will be able to:**

- Compare AI with human intelligence and traditional information processing and discuss its strengths and limitations as well as its application to complex and human-centred problems.
- Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems.
- Analyze the structures and algorithms of a selection of techniques related to searching, reasoning, machine learning, and language processing.
- Design AI functions and components involved in intelligent systems such as computer games, expert systems, semantic web, information retrieval, machine translation, mobile robots, decision support systems, and intelligent tutoring systems.

**UNIT 1: INTRODUCTION TO R AND ARTIFICIAL INTELLIGENCE:**

The importance of data, learning statistics and tools, Math for Data Analysis: Brush up your knowledge of Linear Algebra, Matrices, basic statistics and their application for Data Analysis, how to apply statistics tools to transform and apply them for business context, understanding importance of data analytics to real life problems. Basics of R, Fundamentals of R programming language.Basics of Big Data: Understand the big data ecosystem, Understand the Hadoop Ecosystem architecture, Understanding data - HDFS, Hive, HBase, Spark, Understanding ETL - Flume, Sqoop and other tools. Basic idea of Machine Learning and Artificial Intelligence: Gain an understanding of how ML/AI helps organizations achieve their business success, Build a foundation for using ML/AI , Learn real-world case studies of the ML/AI – Email filters, recommendations, social media recommendations.

## UNIT 2: BASIC OPERATIONS ON DATA:

Statistics : Statistical Modeling, Regression, Analysis of Variance, Analysis of Covariance, Generalized Linear Models. Data Wrangling : importing data, data cleansing, data manipulations, quick data transformations, slicing & dicing the data. Data Mining : Working with R data: vectors, matrices, arrays, and data frames, Vectorized Operations, Control Structures, Functions, Scoping Rules of R, Coding Standards for R, Loop functions, Debugging, Profiling, Simulation. Data Modelling : Data distributions, Model fitting, Model selection, Model validation, Model deployment, Statistical learning, Linear regression, Classification techniques. Data Visualization : Visualization techniques using ggplot2 which is one of the best data visualization tools around.

## UNIT 3: ADVANCED OPERATIONS ON DATA:

Statistics : Generalized Additive Models, Mixed-Effects Models, Non-linear Regression, Tree Models, Time Series Analysis, Multivariate Statistics, Spatial Statistics, Survival Analysis. Data Mining and Wrangling : Advanced imputation techniques using Machine learning. Data Modelling : Resampling methods, Linear model selection and Regularization, Moving beyond Linearity, Tree based methods, Support Vector Machines, Unsupervised Learnings.

## UNIT 4: NATURAL LANGUAGE PROCESSING:

Basics of text processing in R, Extract features from unstructured text and build machine learning models on text data, Conduct sentiment analysis, learn to parse English sentences and extract meaning from them, Explore the applications of text analytics in new areas and various business domains.

## UNIT 5: DEEP LEARNING AND BIG DATA TECHNIQUES:

Neural Net, Advanced MC Learning : Understand the components and structure of artificial neural networks, Techniques used to train highly complex neural networks, Applications in text analytics, Creating and deploying networks using Tensorflow and keras. Reinforcement Learning : Understanding the basics of Reinforcement Learning and its applications, Model processes as Markov chains, learn algorithms for solving optimisation problems, Learn Q-learning algorithms to solve complex Reinforcement Learning problems. Big data : Performing AI on big data using H2O, Spark.

Projects on AI.

**REFERENCES:**

1. Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (First Edition)
2. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction

# Six Phrase – MySlate (Learning Management System)

## Pointers:

- ❖ Six Phrase will be offering our LMS (Learning Management System) to all the students undergoing training. Contents of the LMS are listed below for reference.
- ❖ Video Lectures, Practice Modules and Assessments will be available to the students as part of the LMS.
- ❖ Six Phrase will be conducting periodic Assessments as part of this training program and reports and performance analysis will be shared with the college management on regular intervals.

## Target Outcome:

- ✓ Continuous Performance Monitoring using **https://www.stopstalk.com** & **http://sixphrase.com.**
- ✓ Outcome based training with the below mentioned target scores
  - **Marquee Students >25000 points**
  - **Super Dream Students > 20000 points**
  - **Dream Students >18000 points**
  - **Service Students >12000 points**

## Online Test Portal:

- ✓ One year access to 600+ Online Tests (Aptitude, Technical, **150+ Company Specific Tests)** using My Slate.

# Six Phrase – MySlate – Technical Training Syllabus

- ✓ **Technical:**
  Basic C – 330 Practice Programs & 15 Video Lectures
  Advanced C – 330 Practice Programs & 30 Video Lectures
  Basic DS – 300 Practice Programs & 61 Video Lectures
  Advanced DS – 300 Practice Programs & 30 Video Lectures
  Algorithms – 300 Practice Programs & 84 Video Lectures
  Java – 330 Practice Programs & 52 Video Lectures
  Python – 330 Practice Programs & 30 Video Lectures

- ✓ **Aptitude:**
  Quantitative Aptitude – 120 Tests & 35 Video Lectures
  Reasoning Aptitude – 120 Tests & 25 Video Lectures
  Verbal Aptitude – 90 Tests & 15 Video Lectures

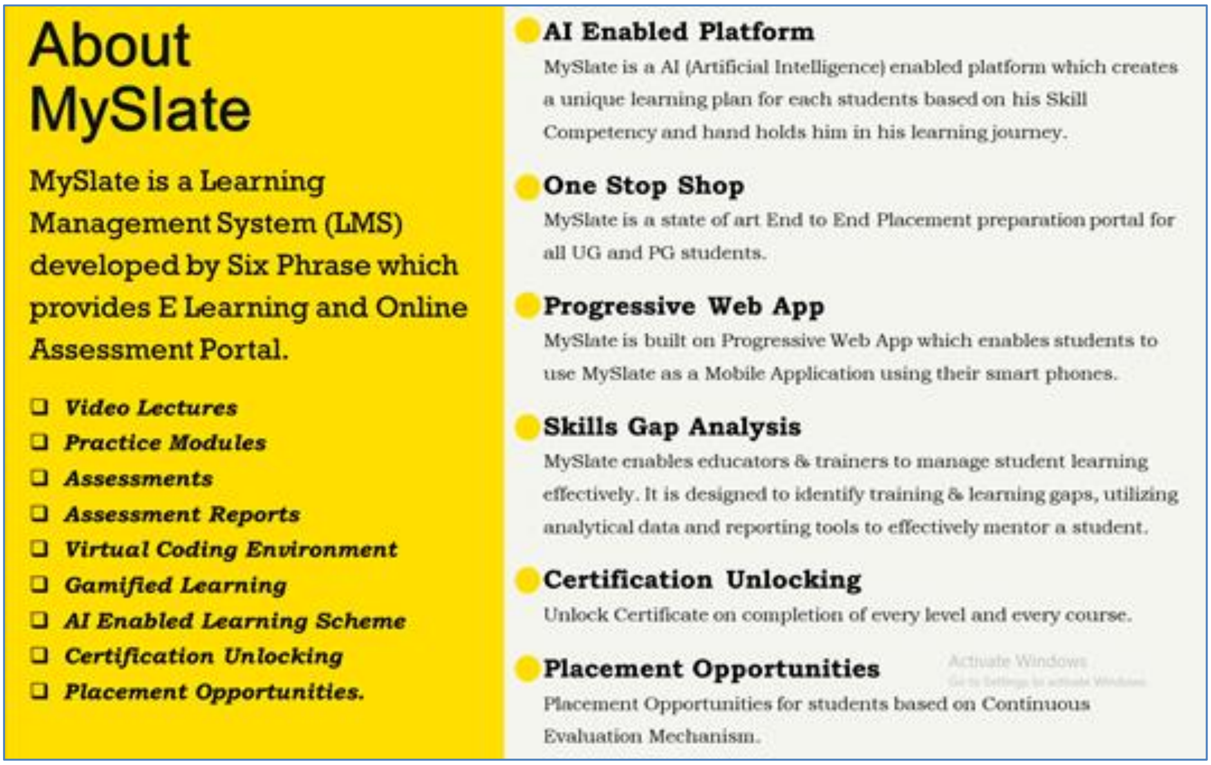
- ✓ **Company Specific Tests:**
  150+ Product & Service Companies Specific Tests & Video Lectures

- ✓ **Competitive Programming:**
  Google Code Jam – 10 Tests & 27 Video Lectures
  TCS CodeVita - 20 Tests & 27 Video Lectures
  Infy TQ - 20 Tests & 15 Video Lectures
  Hack with Infy - 10 Tests & 10 Video Lectures

INTRODUCING MYSLATE

SixPhrase - MySlate - South India's Largest Employability Training, Skill Development & Career Enhancement Enterprise.

# Six Phrase – MySlate – Technical Training Syllabus



## About MySlate

MySlate is a Learning Management System (LMS) developed by Six Phrase which provides E Learning and Online Assessment Portal.

- ❑ ***Video Lectures***
- ❑ ***Practice Modules***
- ❑ ***Assessments***
- ❑ ***Assessment Reports***
- ❑ ***Virtual Coding Environment***
- ❑ ***Gamified Learning***
- ❑ ***AI Enabled Learning Scheme***
- ❑ ***Certification Unlocking***
- ❑ ***Placement Opportunities.***

**AI Enabled Platform**

MySlate is a AI (Artificial Intelligence) enabled platform which creates a unique learning plan for each students based on his Skill Competency and hand holds him in his learning journey.

**One Stop Shop**

MySlate is a state of art End to End Placement preparation portal for all UG and PG students.

**Progressive Web App**

MySlate is built on Progressive Web App which enables students to use MySlate as a Mobile Application using their smart phones.

**Skills Gap Analysis**

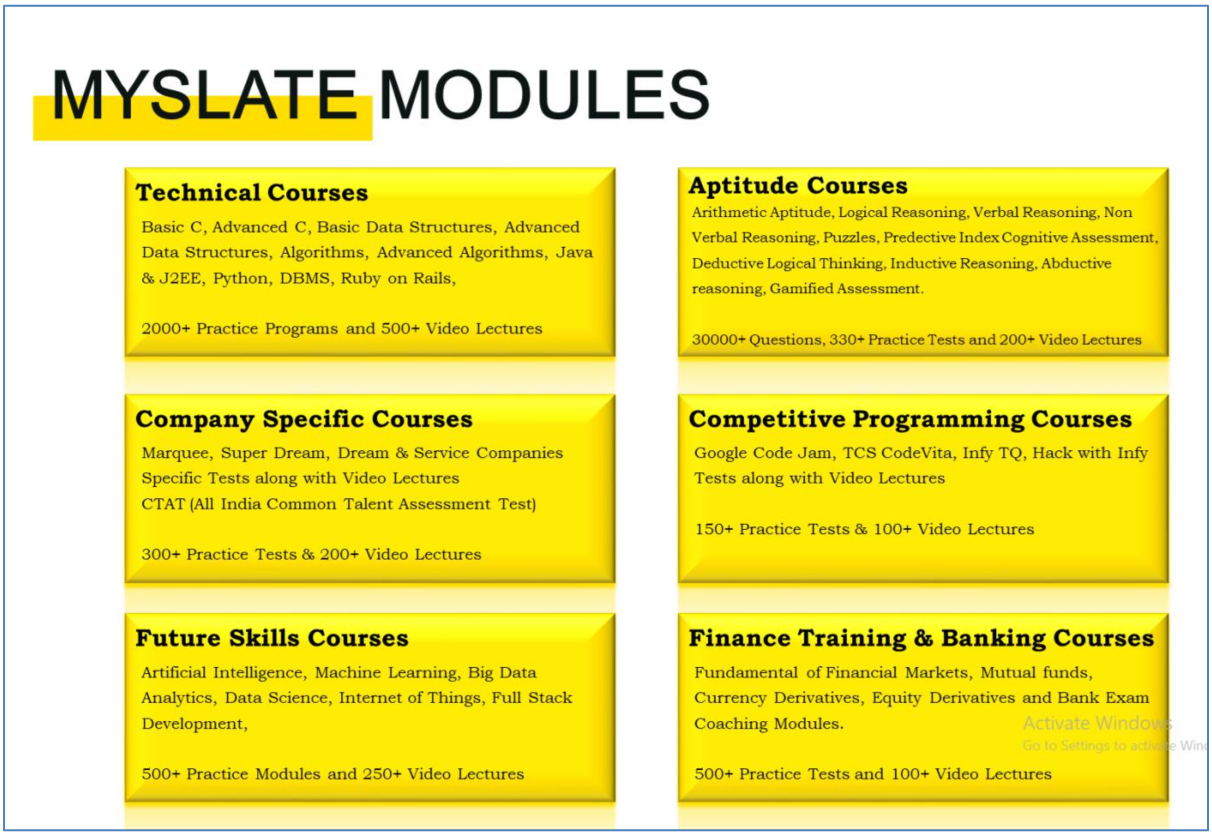
MySlate enables educators & trainers to manage student learning effectively. It is designed to identify training & learning gaps, utilizing analytical data and reporting tools to effectively mentor a student.

**Certification Unlocking**

Unlock Certificate on completion of every level and every course.

**Placement Opportunities**

Placement Opportunities for students based on Continuous Evaluation Mechanism.



## MYSLATE MODULES

**Technical Courses**

Basic C, Advanced C, Basic Data Structures, Advanced Data Structures, Algorithms, Advanced Algorithms, Java & J2EE, Python, DBMS, Ruby on Rails,

2000+ Practice Programs and 500+ Video Lectures

**Aptitude Courses**

Arithmetic Aptitude, Logical Reasoning, Verbal Reasoning, Non Verbal Reasoning, Puzzles, Predective Index Cognitive Assessment, Deductive Logical Thinking, Inductive Reasoning, Abductive reasoning, Gamified Assessment.

30000+ Questions, 330+ Practice Tests and 200+ Video Lectures

**Company Specific Courses**

Marquee, Super Dream, Dream & Service Companies Specific Tests along with Video Lectures

CTAT (All India Common Talent Assessment Test)

300+ Practice Tests & 200+ Video Lectures

**Competitive Programming Courses**

Google Code Jam, TCS CodeVita, Infy TQ, Hack with Infy Tests along with Video Lectures

150+ Practice Tests & 100+ Video Lectures

**Future Skills Courses**

Artificial Intelligence, Machine Learning, Big Data Analytics, Data Science, Internet of Things, Full Stack Development,

500+ Practice Modules and 250+ Video Lectures

**Finance Training & Banking Courses**

Fundamental of Financial Markets, Mutual funds, Currency Derivatives, Equity Derivatives and Bank Exam Coaching Modules.

500+ Practice Tests and 100+ Video Lectures