

OBJECTIVE

It explores a range of modern programming languages and programming techniques and performs comparison of the same, and then gives the idea of evaluating programming language features critically with respect to the way they support good software engineering practice. Also, it describes analyzing of solving a problem in terms of several programming paradigms, so that one can determine the advantages and disadvantages of each approach.

UNIT – I INTRODUCTION 9

The art of Language design – Programming language spectrum - Compilation and Interpretation – Evaluation of Programming languages – Syntax and Semantics of language Clite - Names – Types – Type Systems - Binding – Scope – Static – Dynamic – Abstract Data types

UNIT – II SEMANTICS 9

Expression – Assignment - Control flow – Input/output – exception handling – state transformation – partial functions – semantics with dynamic typing – Formal treatment of semantics

UNIT – III FUNCTIONS 9

Call and Return – Parameter passing – function declaration – semantics of call and return – formal treatment of types and semantics – memory management – dynamic arrays – garbage collection

UNIT – IV PROGRAMMING TECHNIQUES 9

Imperative programming – C – ADA – Perl – Object Oriented Programming – Small Talk- Java – Python – Functional Programming – Scheme – Haskell

UNIT – V MODERN PROGRAMMING TECHNIQUES 9

Logic programming – prolog – Event-Driven programming – Concurrent Programming – Concepts – Synchronization strategies – Language level mechanism - Interprocess communication – Scripting languages.

TOTAL: 45**TEXTBOOKS:**

1. Allen B. Tucker and Robert E. Noonan, Programming Languages - Principles and Paradigms, Second Edition, Tata McGraw Hill, 2009

REFERENCES:

1. Robert W. Sebesta, Concepts of Programming Languages, Sixth Edition, Addison Wesley, July 24,2003.
2. Michael L Scott, Programming Language Pragmatics, Third Edition, Morgan Kauffman, 2009