

**2/4 B.Tech. SECOND SEMESTER  
PRINCIPLES OF PROGRAMMING LANGUAGES**

**CS4T2**

**Required**

**Credits: 4**

**Lecture: 4 periods/week**

**Internal assessment: 30 marks**

**Tutorial: 1 period /week**

**Semester end examination: 70 marks**

---

**Course context and Overview:** To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming

---

**Prerequisites: Familiarity with programming in general.**

**Objectives:**

This course teaches the principles of programming languages and basic skills needed to understand and analyze syntax for problem solving using different paradigms. The following programming paradigms will be covered: logic, functional, procedural, imperative and object-oriented. The course describes the fundamental concepts of programming languages such as Pascal, C, C++, Lisp, Prolog, Smalltalk etc., by discussing the design issues of the various languages constructs, examining the design choices of these constructs in some of the common languages.

The main goals of this course are

1. Increased capacity to express ideas.
2. Improved knowledge in choosing appropriate languages.
3. Increased ability to learn new languages.
4. Better understanding of the significance of implementation.
5. Overall advancement of computing.
6. Understand the differences between different programming paradigms.
7. Increased capacity to develop programs in different programming languages.

**Learning Outcomes:**

Ability to:

1. Design parse trees for syntax and semantics of programming languages.
2. Understand characteristics and features of various data types and control structures in programming languages
3. Illustrate the modularity in programs with different parameter passing techniques
4. Use the concept of Abstract data type, concurrency and exception handling in programming
5. Describe the features of logical & functional programming languages

## **UNIT I**

### **Preliminary Concepts:**

Reasons for studying concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Language Implementation – Compilation and Virtual Machines, programming environments.

## **UNIT II**

### **Syntax and Semantics:**

General Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, dynamic semantics

## **UNIT III**

### **Data types:**

Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization. **Prasad V. Potluri Siddhartha**  
**Institute of Technology, Kanuru, Vijayawada.**

## **UNIT IV**

### **Expressions and Statements:**

Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, and guarded commands.

## **UNIT-V**

### **Subprograms and Blocks:**

Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.

## **UNIT VI**

**Abstract Data types:** Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java,

**Concurrency:** Subprogram level concurrency, semaphores, monitors, message passing, Javathreads, C# threads, Ada.

## **UNIT VII**

**Exception handling:** Exceptions, exception propagation, exception handler in C++ and Java, Ada.

**Logic Programming Language:** Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

## **UNIT VIII**

**Functional Programming Languages:** Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages.

## Learning Resources

### TEXT BOOKS:

1. Concepts of Programming Languages Robert .W. Sebesta 6/e, Pearson Education.
2. Programming Languages –Louden, Second Edition, Thomson.

### REFERENCES:

1. Programming languages –Ghezzi, 3/e, John Wiley.
2. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education.
3. Programming languages –Watt, Wiley Dreamtech.