



CS 253

## PRINCIPLES OF PROGRAMMING LANGUAGES

### Course objectives:

- To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
- To introduce notations to describe syntax and semantics of programming languages.
- To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
- To introduce the concepts of ADT and object oriented programming for large scale software development.
- To introduce the concepts of concurrency control and exception handling.

### UNIT I :

**Preliminary Concepts:** Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-offs, Implementation Methods, Programming Environments, Evolution of the Major Programming Languages

**Describing Syntax and Semantics:** General Problem of Describing Syntax, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meaning of Programs.

### UNIT II :

**Names, Binding, Type Checking, and Scopes:** Names, Variables, The Concept of Binding, Type Checking, Strong Typing, Type Compatibility, Scope, Scope and Lifetime, Referencing Environments, Named Constants.

**Data Types:** Primitive Data Types, Character String Types, User- Defined Ordinal Types, Array Types , Associative Arrays, Record Types, Union Types, Pointer and Reference Types.

**Expressions and Assignment Statements:** Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed- Mode Assignment.

## UNIT III

**Statement-Level Control Structures:** Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

**Subprograms:** Fundamentals and Design Issues for Subprograms, Local Referencing Environments, Parameter –Passing Methods, Parameters That are Subprograms Names, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User-Defined Overloaded Operators.

**Implementing Subprograms:** The General Semantics of Calls and Returns, Implementing “Simple” Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping.

**Abstract Data Types:** The Concept of Abstraction, Introduction to Data Abstraction, Design Issues for Abstract Data Types, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulation.

## UNIT IV

**Object Oriented Programming:** Design Issues, Object Oriented Programming in Smalltalk, C++, Java, C#, Ada 95, Ruby, The Object Model of JavaScript, Implementation of Object Oriented Constructs.

**Concurrency:** Subprogram level Concurrency, Semaphores, Monitors, Message Passing, Ada Support for Concurrency, Java Threads, C# Threads, Statement-Level Concurrency.

**Exception Handling and Event Handling:** Introduction to Exception Handling, Exception Handling in Ada, C++ and Java, Introduction to Event Handling, Event Handling with Java.

## UNIT V :

**Functional Programming Languages:** Introduction, Mathematical Functions, Fundamentals of FPL, LISP, Introduction to Scheme, COMMON LISP, ML, Haskell, Application of Functional Programming Languages and A Comparison of Functional and Imperative Languages.

**Logic Programming Languages:** Introduction to Predicate Calculus, Predicate Calculus and Proving Theorems, An Overview of Logic Programming. The Origins, Basic Elements and Deficiencies of Prolog, Applications of Logic Programming.

**Scripting Languages:** Key concepts, Case Study: Python(From the Suggested Reading 2).

**Suggested Reading:**

1. Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education, 2008.
2. Programming languages –Watt, Wiley Dreamtech, 2004.
3. Programming Languages –Louden, Second Edition, Cengage, 2003.
4. Programming languages –Ghezzi, 3/e, John Wiley, 1998.
5. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education, 2001.