

CS 207: PRINCIPLES OF PROGRAMMING LANGUAGES (3-0-2: 4)

Introduction - What is a programming language, Abstractions in programming languages, Computational paradigms, Language definition, Language translation, Language design; A chronology of development.

Language Design Principles - Design criteria, Efficiency, Regularity, C++: A Case study in language design.

Syntax - Lexical structure, Context-free grammars and BNFs, Parse trees and Abstract syntax trees, Ambiguity, Associativity and precedence, EBNFs and syntax diagrams, Parsing techniques and tools, Lexical vs. Syntax vs. Semantics.

Basic Semantics - Attributes, binding and semantic functions, Declarations, blocks and scope, The symbol table, Name resolution and overloading, Allocation, Lifetimes and the environment, Variables and Constants, Aliases, Dangling references and garbage.

Data Types - Data types and type information, Simple types, Type constructors, Type equivalence, Type Checking, Type conversion, Polymorphic type checking, Explicit polymorphism.

Expressions and Statements - Expressions, Conditional Statements and Guards, Loops and Variation on “while”, the “goto” controversy, Exception handling.

Procedures and Environments - Procedure definition and activation, Procedure semantics, Parameter passing mechanisms, Procedure environments, activations and allocation, Dynamic memory management, Exception handling and environments.

Abstract data types and Modules - The algebraic specification of abstract data types, Abstract data type mechanisms and modules, Separate compilation in C, C++ namespaces and Java packages, Modules in earlier languages, Problems with abstract data type mechanisms, The mathematics of abstract data types.

Example features of Object-Oriented, Functional, Logic Programming Languages

Suggested Laboratory Assignments:

1. Use of tools for lexical analysis and parsing,
2. Representation of abstract syntax
3. Prolog programs for recursion, arithmetic operators & relational operators, list processing, string manipulation, compound objects, dynamic database
4. C++ and Java programs for designing classes & objects, constructors & destructors, operator overloading, inheritance, and polymorphism, file operations.

Text Book:

1. Kenneth Loudon, Programming Languages-Principles and Practices, Thomson.

References:

1. Terrence W. Pratt, Programming Languages: Design and Implementation, Pearson.