CSE 301 DESIGN AND IMPLEMENTATION OF PROGRAMMING LANGUAGES

[3 1 0 4]

Prerequisite: CSE101/CSE102, CSE210 **Number of Contact Hours - 48**

Course Objectives:

At the end of the program the student must be able to

- Know about different computational paradigms.
- Understand the language design principles.
- Construct parse trees and abstract syntax trees.
- Distinguish between lexical, syntactical and semantic constructs.
- Choose the right paradigm to solve a given problem.
- Write programs in different program paradigms.
- Write the syntax and semantic of a small language.

1 INTRODUCTION:

What is a Programming Language?, Abstractions in Programming Languages, Computational Paradigms, Language Definition, Language Translation
(Chapter 1 from Text Book sections 1.1 to 1.5) [03]

2 LANGUAGE DESIGN PRINCIPLES

History and Design Criteria, Efficiency, Regularity, Further Language Design Principles
(Chapter 2 from Text Book sections 2.1 to 2.5) [02]

3 SYNTAX

Lexical Structure of Programming Languages, Context-Free Grammars and BNFs Parse Trees and Abstract Syntax Trees, Ambiguity, Associativity, and Precedence EBNFs and Syntax Diagrams, Parsing Techniques and Tools, Lexics Versus Syntax Versus Semantics
(Chapter 6 from Text Book sections 6.1 to 6.7) [06]

4 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
(Chapter 7 from Text Book sections 7.1 to 7.7) [08]

5 DATA TYPES

Data Types and Type Information, Simple Types, Type Constructors
(Chapter 8 from Text Book sections 8.1 to 8.3) [02]

6 EXPRESSIONS AND STATEMENTS

Expressions, Conditional Statements and Guards, Loops and Variation on WHILE, The GOTO Controversy, Exception Handling
(Chapter 9 from Text Book sections 9.1 to 9.5) [04]

7 OBJECT-ORIENTED PROGRAMMING

Software Reuse and Independence, Java: Objects, Classes, and Methods, Inheritance Dynamic Binding
(Chapter 5 from Text Book sections 5.1 and 5.3) [04]

8 FUNCTIONAL PROGRAMMING

Programs as Functions, Functional Programming in an Imperative Language, Scheme: A Dialect of LISP
(Chapter 3 from Text Book sections 3.1 to 3.2) [04]

9 LOGIC PROGRAMMING

Logic and Logic Programs, Horn Clauses, Resolution and Unification
(Chapter 4 from Text Book sections 4.1 to 4.3) [03]

10 FORMAL SEMANTICS

A Sample Small Language, Operational Semantics, Denotational Semantics, Axiomatic Semantics, Proofs of Program Correctness
(Chapter 12 from Text Book sections 12.1 to 12.5) [05]

11 PARALLEL PROGRAMMING

Introduction of Parallel Processing, Parallel Processing and Programming Languages, Threads, Semaphores, Monitors, Message Passing, Parallelism in Non-imperative Languages
(Chapter 13 from Text Book sections 13.1 to 13.7) [07]

Text Books:

1. Kenneth C. Louden,Kenneth A. Lambert, Programming Languages – Principles and Practice, 3rd Edition, Cengage Learning 2012.

References:

1. Harold Abelson, Gerald Jay Sussman and Julie Sussman, "Structure and Interpretation of Computer Programs", 2nd Edition, MIT Press, 1996, available online at http://mitpress.mit.edu/sicp/full-text/book/book.html
2. Robert W. Sebesta, "Concepts of Programming Languages", Ninth Edition, Addison-Wesley, 2010
3. Michael L. Scott, "Programming Language Pragmatics", Third edition, Morgan Kaufmann Publishers, 2009.