PRINCIPLES OF PROGRAMMING LANGUAGES

Course Code:13CS1106

L	T	P	C
4	1	0	3

Course Educational Objectives:

To enable the students to learn about various constructs and their respective comparisons in different high-level languages so that he can choose suitable programming language for solving a particular problem.

Course Outcomes:

Upon completion of this course, the student should be able to:

- Understand the reason for learning a programming language.
- Understand the usage and existence of various programming languages.
- Understand the various features of a programming language with syntactical description.
- Describe the usage of data types in various languages.
- Understand the concepts like abstract data types, sub programs and will be able to apply in realistic manner.
- Understand the overview of programming language like Prolog, FPL, LISP, ML.

UNIT-I (12 Lectures)

PRELIMINARY CONCEPTS:

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

methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute SYNTAX

AND SEMANTICS: General Problem of describing Syntax and Semantics, formal grammars.

UNIT-II (12 Lectures)

DATA TYPES:

Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization.

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.

UNIT-III (12 Lectures)

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

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, C#, Ada 95

UNIT-IV (12 Lectures)

CONCURRENCY:

Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.

EXCEPTION HANDLING:

Exceptions, exception Propagation, Exception handler in Ada, C++ and Java.

UNIT-V (12 Lectures)

FUNCTIONAL PROGRAMMING LANGUAGES:

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

LOGIC PROGRAMMING LANGUAGE:

Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

TEXT BOOKS :

1. Robert .W. Sebesta, *"Concepts of Programming Languages"*, 6th Edition , Pearson Education, 2002 .(1 to 5 units)
2. Louden, *"Programming Languages"*, 3rd Edition,2012.

REFERENCES:

1. Ghezzi, *"Programming Languages"*, 3rd Edition, John Wiley, 2008.

2 Pratt and Zelkowitz, *"Programming Languages Design and Implementation"*, 4th Edition PHI/Pearson Education,2008.

3. Watt, "*Programming languages*", 1st Edition, Wiley Dreamtech., 2004.
4. Patric Henry Winston and Paul Horn, *"LISP"*,3rd Edition, Wiley Dreamtech,2005.
5. Clocksin, *"Programming in PROLOG"*, 5th Edition, Springer, 2004.