



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SYLLABUS

PRINCIPLES OF PROGRAMMING LANGUAGES

B. Tech. III Year I Semester								
Course Code	Category	Hours/Week			Credits	Maximum Marks		
23CS508	Core	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100
Contact Classes:60	Tutorial Classes: NIL	Practical Classes: NIL			Total Classes:60			
Prerequisites: 1) A course on “Programming for problem solving” 2) A course on “Data Structures”								

Course Objectives

1. Introduce important paradigms of programming languages
2. To provide conceptual understanding of high-level language design and implementation
3. Topics include programming paradigms; syntax and semantics; data types, expressions and statements; sub programs and blocks; abstract data types; concurrency; functional and logic programming languages; and scripting languages.

Course Outcomes

1. Acquire the skills for expressing syntax and semantics informal notation
2. Identify and apply a suitable programming paradigm for a given computing application
3. Gain knowledge of and able to compare the features of various programming languages.

COURSE SYLLABUS

UNIT- I

Preliminary Concepts: Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, and language design trade-offs, implementation methods, programming environments, Evolution of Major Programming Languages.

Syntax and Semantics: General problem of describing syntax, formal methods of describing syntax, attribute grammars, describing the meanings of programs.

UNIT- II

Names, Bindings, and Scopes: Introduction, names, variables, concept of binding, scope, scope and lifetime, referencing environments, named constants. Data types: Introduction, primitive, character, string types, user defined ordinal types, array, associative arrays, record, tuple types, list types, union types, pointer and reference types, type checking, strong typing, type equivalence

Expressions and Statements: Arithmetic expressions, overloaded operators, type conversions, relational and Boolean expressions, short- circuit evaluation, assignment statements, mixed-

mode assignment Control Structures – introduction, selection statements, and iterative statements, unconditional branching, guarded commands.

UNIT - III

Subprograms: Fundamentals of subprograms, design issues for subprograms, local referencing environments, and parameter passing methods, parameters that are subprograms, calling subprograms indirectly, overloaded subprograms, generic subprograms, design issues for functions, and user-defined overloaded operators, closures, co routines.

Implementing subprograms: 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, introductions to data abstraction, design issues, language examples, parameterized ADT, encapsulation constructs, naming encapsulations.

UNIT- IV

Object Oriented Programming: Design issues for OOP, OOP in Smalltalk, C++, Java, Ada 95, Ruby, Implementation of Object-Oriented constructs.

Concurrency: introduction, introduction to subprogram level concurrency, semaphores, monitors, message passing, Ada support for concurrency, Java threads, concurrency in functional languages, statement level concurrency.

Exception Handling and Event Handling: Introduction, exception handling in Ada, C++, Java, introduction to event handling, event handling with Java and C#.

UNIT - V

Functional Programming Languages: Introduction, mathematical functions, fundamentals of functional programming language, LISP, support for functional programming in primarily imperative languages, comparison of functional and imperative languages.

Logic Programming Language: Introduction, an overview of logic programming, basic elements of prolog, deficiencies of prolog, applications of logic programming.

Scripting Language: Pragmatics, Key Concepts, Case study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

TEXT BOOK:

1. Concepts of Programming Languages, Robert .W. Sebesta 10th edition, Pearson Education.
2. Programming Language Design Concepts, D. A. Watt, Wiley India Edition.

REFERENCE BOOKS:

1. Programming Languages, A.B. Tucker, R.E. Noonan, TMH.
2. Programming Languages, K. C. Loudon and K A Lambert., 3rd edition, Cengage Learning.
3. Programming Language Concepts, C Ghezzi and M Jazayeri, Wiley India.
4. Programming Languages 2nd Edition Ravi Sethi Pearson.
5. Introduction to Programming Languages Arvind Kumar Bansal CRC Press.