CP5015 PRINCIPLES OF PROGRAMMING LANGUAGES

DETAILED SYLLABUS

OBJECTIVES

- To understand and describe syntax and semantics of programming languages.
- To understand Data, Data types, and Bindings.
- To learn the concepts of functional and logical programming.
- To explore the knowledge about concurrent Programming paradigms.

UNIT I ELEMENTS OF PROGRAMMING LANGUAGES

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. Describing Syntax and Semantics -Introduction - The General Problem of Describing Syntax-Formal Methods of Describing Syntax - Attribute Grammars - Describing the Meanings of Programs: Dynamic Semantics.

UNIT II DATA TYPES-ABSTRACTION

Introduction - Primitive Data Types- Character String Types- User-Defined Ordinal TypesArray types- Associative Arrays-Record Types- Tuple Types-List Types -Union Types - Pointer and Reference Types -Type Checking- Strong Typing -Type Equivalence - Theory and Data Types-Variables-The Concept of Binding -Scope - Scope and Lifetime - Referencing Environments - Named Constants- The Concept of Abstraction- Parameterized Abstract Data Types- Encapsulation Constructs- Naming Encapsulations

UNIT III FUNCTIONAL PROGRAMMING

Introduction- Mathematical Functions- Fundamentals of Functional Programming Languages- The First Functional Programming Language: LISP- An Introduction to SchemeCommon LISP- Haskell-F# - ML : Implicit Types- Data Types- Exception Handling in ML. Functional Programming with Lists- Scheme, a Dialect of Lisp- The Structure of Lists- List Manipulation- A Motivating Example: Differentiation- Simplification of Expressions- Storage Allocation for Lists.

UNIT IV LOGIC PROGRAMMING

Relational Logic Programming- Syntax- Basics- Facts- Rules- Syntax- Operational Semantics- Relational logic programs and SQL operations- Logic Programming- SyntaxOperational semantics- Data Structures-Meta-tools: Backtracking optimization (cuts); Unify; Meta-circular interpreters- The Origins of Prolog- Elements- of Prolog-Deficiencies of Prolog Applications of Logic Programming.

UNIT V CONCURRENT PROGRAMMING

Parallelism in Hardware- Streams: Implicit Synchronization-Concurrency as Interleaving Liveness Properties- Safe Access to Shared Data- Concurrency in Ada- Synchronized Access to Shared Variables- Synthesized Attributes- Attribute Grammars- Natural Semantics- Denotational Semantics -A Calculator in Scheme-Lexically Scoped Lambda Expressions- An Interpreter-Recursive Functions.

REFERENCES

1. Ghezzi, “Programming Languages”, 3rd Edition, John Wiley, 2008
2. John C. Mitchell, “Concepts in Programming Languages”, Cambridge University Press, 2004.
3. Louden, “Programming Languages”, 3rd Edition, 2012.
4. Ravi Sethi, “Programming Languages: Concepts and Constructs”, 2nd Edition, Addison Wesley, 1996.
5. Robert .W. Sebesta, “Concepts of Programming Languages”, 10th Edition, Pearson Education, 2002.