INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

PROGRAMMING LANGUAGE PARADIGMS								
V Semester: CSE | CSE(CS)								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
ACSD22	Elective	3	0	0	3	40	60	100
Contact Classes: 48	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 48**		
Prerequisite: There is no prerequisite to take this course								

I. COURSE OVERVIEW:

This course explores the theoretical foundations and practical applications of diverse programming paradigms. Students will investigate the historical evolution of programming languages and understand their design considerations. The course covers functional programming using languages like Scheme, ML, and Haskell; logical programming through Prolog and Curry; object-oriented programming via Smalltalk, Java, and C++; and parallel programming concepts such as threads, semaphores, and message passing. Emphasis is placed on the differences in computational models, abstraction mechanisms, and implementation strategies. By the end of the course, students will be equipped with the knowledge to analyze, compare, and apply appropriate paradigms to solve various computational problems efficiently.

II. COURSE OBJECTIVES:

The students will try to learn:

- **I.** The historical development, design principles, and classification of programming languages, along with their paradigms and translation mechanisms.
- **II** The principles of functional, logical, object-oriented, and parallel programming languages through practical examples.
- **III** The ability to critically compare paradigms and select suitable programming models based on problem requirements and language features.

III.COURSE OUTCOMES:

At the end of the course students should be able to:

- CO1 Explain the historical context, design considerations, and classification of programming languages, including abstraction and computational paradigms.
- CO2 Demonstrate the functional programming concepts using languages like Scheme, ML, and Haskell, including lambda calculus and lazy evaluation.
- CO3 Apply logical programming principles using Prolog and analyze the semantics of logic-based languages using resolution and unification.
- CO4 Compare object-oriented programming principles using languages like Java, Smalltalk, and C++, addressing reuse, encapsulation, and inheritance.
- CO5 Understand the fundamental parallel programming constructs such as threads, semaphores, monitors, and message passing.
- CO6 Evaluate the strengths and limitations of various programming paradigms and apply them appropriately to solve computational problems.

IV.COURSE CONTENT:

MODULE –I: INTRODUCTION (10)

Origins of programming language, abstraction, computational paradigms, language definitions and translation and the future of programming language.

Programming Language Design: History, efficiency, regularity, security, extensibility, C++: An object-oriented extension of C, Python: A general – purpose scripting language.

MODULE –II: FUNCTIONAL PROGRAMMING (10)

Programs as functions, Scheme: A dialect of Lisp, ML: Functional Programming with static typing, Delayed evaluation, Haskell- A fully carried lazy language with overloading, The mathematics of functional programming: Lambda calculus.

MODULE –III: LOGICAL PROGRAMMING (09)

Logic and Logic programs, Horn Clauses, Resolution and unification, The Language Prolog.

Problems with logic programming, Curry: A functional logic language.

MODULE –IV: OBJECT-ORIENTED PROGRAMMING (09)

Software reuse and independence, Smalltalk, Java, C++, Design issues in object-oriented languages, Implementation issues in object-oriented languages.

MODULE –V: PARALLEL PROGRAMMING (10)

Introduction to Parallel Processing, Parallel Processing and Programming language, Threads, Semaphores, Monitors, Message Passing, Parallelism in non-imperative languages.

V. TEXT BOOKS:

1. KC Louden, “Programming Language Principles and Practice”, Thomson course technology. 2007.
2. Robert W. Sebesta, “Concepts of Programming Languages”, Pearson Education, ISBN:9789356067417.
3. Maurizio Gabbrielli, Simone Martini, “Programming Languages: Principles and Paradigms”, Springer, 2023.

VI. REFERENCE BOOKS:

1. Alfred V. Aho, Ravi Sethi. “Compilers Principles, Techniques and Tools”, Pearson education, 2nd Edition, 2007.
2. David Anthony Watt, “Programming Language Concepts and Paradigms”, ISBN: 0137288743.

VII. ELECTRONICS RESOURCES

1. https://www.google.com/search?q=video+links+for+programming+language+paradigms&sca_esv=137e48d954af
2. https://www.youtube.com/watch?v=mB4u4ETsn28
3. https://www.globalnerdy.com/2019/12/10/worth-watching-videos-on-programming-paradigms-and-object-oriented-vs-functional-programming/

VIII. MATERIALS ONLINE:

1. Course template
2. Tutorial question bank
3. Tech-talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)