	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									
CO2	✓	✓	✓	✓	✓				✓			✓
CO3	✓	✓	✓	✓	✓				✓		✓	✓
CO4	✓	✓	✓	✓	✓				✓			✓
CO5	✓	✓	✓	✓	✓				✓		✓	✓

CS6302 PROGRAMMING PARADIGMS

Prerequisites for the course: Data Structures and Algorithms

OBJECTIVES:

- To introduce the major programming paradigms with the principles and the techniques involved in the design and implementation of modern programming languages
- To introduce the framework for specifying and reasoning about programming languages
- To analyse a given program from the perspective of good programming practices
- To compare and contrast the range of programming paradigms
- To evaluate programming language features critically with respect to the way they support good software engineering practices
- To discuss the appropriateness of the use of a given programming paradigm within a given environment

CS6302 PROGRAMMING PARADIGMS	L	T	P	EL	CREDITS
	3	0	0	3	4

OBJECTIVES:

MODULE I :	L	T	P	EL
	3	0	0	5

The art of Language design – Programming language spectrum - Compilation and Interpretation– Evaluation of Programming languages

SUGGESTED ACTIVITIES :

- Activity based learning - brain storming quizzes and puzzles of programming languages

SUGGESTED EVALUATION METHODS:

- Quizzes

MODULE II :	L	T	P	EL
Languages – Syntax and Semantics of language C-lite - Names – Types – Type Systems - Binding – Scope – Static – Dynamic – Abstract Data types	4	0	0	5

SUGGESTED ACTIVITIES :

- Using peer learning- Interaction and group discussion about data types

SUGGESTED EVALUATION METHODS:

- Quizzes
- Assignment problems

MODULE III :	**L**	**T**	**P**	**EL**
Expression – Assignment - Control flow – Input/output – exception handling - exception hierarchy-throwing and catching exception	**4**	**0**	**0**	**5**

SUGGESTED ACTIVITIES :

- Problem based learning for solving problems using various exception handling techniques in the module.

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE IV :	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**6**
Introduction to semantics -state transformation – partial functions – semantics with dynamic typing – Formal treatment of semantics				

SUGGESTED ACTIVITIES :

- Outcome based learning- various assessment tests for the above four modules.

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE V :	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**6**
Functions - Call and Return – Parameter passing – function declaration – semantics of call and return				

SUGGESTED ACTIVITIES :

Activity based learning - quizzes and puzzles related to using functions

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE VI:	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**5**

Formal treatment of types and semantics – memory management – dynamic arrays – garbage collection

SUGGESTED ACTIVITIES :

- Problem based learning - Solving problems using dynamic arrays

SUGGESTED EVALUATION METHODS:

- Assignment problems

MODULE VII	**L**	**T**	**P**	**EL**
	4	**0**	**0**	**5**

Programming techniques-Imperative programming – C – ADA – Perl

SUGGESTED ACTIVITIES :

- Based on project learning, develop a mini project based on C or Perl

SUGGESTED EVALUATION METHODS:

- Assignment problems

MODULE VIII	**L**	**T**	**P**	**EL**
	4	**0**	**0**	**5**

Object Oriented Programming -grouping of data and operations-constructs for program structuring-information hiding-program design with modules - Object Oriented Programming – Small Talk-Java– Python

SUGGESTED ACTIVITIES :

- Case study to understand OOPs concepts of Java and Python

SUGGESTED EVALUATION METHODS:

- Assignment problems

MODULE IX	**L**	**T**	**P**	**EL**
	3	**0**	**0**	**5**

Functional Programming – Introduction to Scheme and Haskell- Expressions-types and functions

SUGGESTED ACTIVITIES :

- Problem solving paradigms in Functional programming

SUGGESTED EVALUATION METHODS:

- Assignment problems

MODULE X	L	T	P	EL
	4	0	0	5
Logic programming – Prolog – Event-Driven programming – Concurrent Programming – Concepts – Synchronization strategies – Language level mechanism - Interprocess communication – Scripting languages.				
SUGGESTED ACTIVITIES : • Project based learning to apply suitable concepts for a small application.				
SUGGESTED EVALUATION METHODS: • Mini Project evaluation				

TEXT BOOKS:

1. Michael L Scott, "Programming Language Pragmatics", Third Edition, Morgan Kauffman, 2009.
2. Allen B. Tucker and Robert E. Noonan, "Programming Languages - Principles and Paradigms", Second Edition, Tata McGraw Hill, 2009.

REFERENCES

1. Daniel P. Friedman and Mitchell Wand, "Essentials of Programming Languages", Third Edition, The MIT Press, 2008.
2. Robert W. Sebesta, "Concepts of Programming Languages", Sixth Edition, Addison Wesley, 2003.
3. Terrence W. Pratt, Marvin V. Zelkowitz, "Programming Languages: Design and Implementation ", 4th Edition, Pearson, 2000.

OUTCOMES:

Upon completion of the course, the students will be able to:

- Write programs related to syntax and semantics
- Compare programs between C, Perl and Small Talk
- Write programs using scripting languages
- Demonstrate event-driven and concurrent programming using Prolog
- Apply Prolog for developing distributed systems

EVALUATION METHOD:

Category of Course	Continuous Assessment	Mid – Semester Assessment	End Semester
Theory	**40**	**20**	**40**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									
CO2	✓	✓	✓	✓	✓				✓			✓
CO3	✓	✓	✓	✓	✓				✓		✓	✓
CO4	✓	✓	✓	✓	✓				✓			✓
CO5	✓	✓	✓	✓	✓				✓		✓	✓

CS6303 DISTRIBUTED SYSTEMS	L	T	P	EL	CREDITS
	3	0	0	3	4

Prerequisites for the course: NONE

OBJECTIVES:

- To understand the foundations of distributed systems
- To learn issues related to clock Synchronization and the need for global state in distributed systems
- To learn distributed mutual exclusion and deadlock detection algorithms
- To understand the significance of agreement, fault tolerance and recovery protocols in distributed systems
- To learn the characteristics of peer-to-peer and distributed shared memory systems

.

MODULE I INTRODUCTION	L	T	P	EL
	4	0	0	3

Definition –Relation to computer system components –Motivation –Relation to parallel systems – Message-passing systems versus shared memory systems –Primitives for distributed communication –Synchronous versus asynchronous executions –Design issues and challenges.

SUGGESTED ACTIVITIES :

- EL – Fundamentals of Distributed Systems
- Flipped classroom and activity

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes

MODULE II A MODEL OF DISTRIBUTED COMPUTATIONS AND LOGICAL TIME	L	T	P	EL
	6	0	0	3

A distributed program –A model of distributed executions –Models of communication networks –Global state –Cuts –Past and future cones of an event –Models of process communications –A framework for a system of logical clocks –Scalar time –Vector time –Physical clock synchronization: NTP.

SUGGESTED ACTIVITIES :

- Flipped classroom and activity
- EL – Basics of Communication Networks

SUGGESTED EVALUATION METHODS:

- Assignment problems
- Quizzes