SEMESTER S7

PROGRAMMING LANGUAGES

(Common to CS/CR/CM/CA/AD/AM)

Course Code	**PECST758**	**CIE Marks**	40
Teaching Hours/Week (L: T:P: R)	3:0:0:0	**ESE Marks**	60
Credits	3	**Exam Hours**	2 Hrs. 30 Min.
Prerequisites (if any)	None	**Course Type**	Theory

Course Objectives:

1. To enable the students understand various constructs and their respective comparisons in different high-level languages so that he can choose a suitable programming language for solving a particular problem
2. To develop the student's ability to understand the salient features and paradigms in the landscape of programming languages.

SYLLABUS

KTU SPECIAL

Module No.	Syllabus Description	Contact Hours
1	Introduction - The Origins of Programming Languages, Abstractions in Programming Languages, Computational Paradigms, Language Definition, Language Translation, The Future of Programming Languages; Language Design Criteria - Historical Overview, Efficiency, Regularity, Security, Extensibility, C++: An Object-Oriented Extension of C, Python: A General-Purpose Scripting Language; Syntax and Analysis Parsing: Lexical Structure of Programming Languages, Context-Free Grammars and BNFs, Parse Trees and Abstract Syntax Trees, Ambiguity, Associativity, and Precedence, EBNFs and Syntax Diagrams, Parsing Techniques and Tools, Lexics vs. Syntax vs. Semantics, Case Study: Building a Syntax Analyzer for TinyAda;	9
2	Basic Semantics- Attributes, Binding, and Semantic Functions, Declarations, Blocks, and Scope, The Symbol Table, Name Resolution and Overloading, Allocation, Lifetimes, and the Environment, Variables and Constants, Aliases, Dangling References, and Garbage, Case Study: Initial Static Semantic Analysis of TinyAda. Data Types - Data Types and Type Information, Simple Types, Type Constructors, Type Nomenclature in Sample Languages, Type Equivalence,	9

	Type Checking, Type Conversion, Polymorphic Type Checking, Explicit Polymorphism, Case Study: Type Checking in TinyAda.	
3	Expressions and Statements - Expressions, Conditional Statements and Guards, Loops and Variations on WHILE, The GOTO Controversy and Loop Exits, Exception Handling, Case Study: Computing the Values of Static Expressions in TinyAda. Procedures and Environments- Procedure Definition and Activation, Procedure Semantics, Parameter-Passing Mechanisms, Procedure Environments, Activations, and Allocation, Dynamic Memory Management, Exception Handling and Environments, Case Study: Processing Parameter Modes in TinyAda.	9
4	Abstract Data Types and Modules- The Algebraic Specification of Abstract Data Types, Abstract Data Type Mechanisms and Modules, Separate Compilation in C, C++ Namespaces, and Java Packages, Ada Packages, Modules in ML, Modules in Earlier Languages, Problems with Abstract Data Type Mechanisms, The Mathematics of Abstract Data Types.	9

Course Assessment Method
(CIE: 40 marks, ESE: 60 marks)

KTU SPECIAL

Continuous Internal Evaluation Marks (CIE):

Attendance	Assignment/ Microproject	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
5	**15**	**10**	**10**	**40**

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
• 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks **(8x3 =24 marks)**	• Each question carries 9 marks. • Two questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 3 subdivisions. **(4x9 = 36 marks)**	**60**

Course Outcomes (COs)

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

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Understand the history of programming languages and introduce abstraction, the concept of different language paradigms, and an overview of language design criteria.	**K1**
CO2	Describe how the syntactic structure of a language can be precisely specified using context-free grammar rules in Backus-Naur form (BNF).	**K2**
CO3	Explain the abstractions of the operations that occur during the translation and execution of programs.	**K2**
CO4	Apply the data types in various languages	**K3**
CO5	Apply procedure activation and parameter passing; and exceptions and exception handling.	**K4**

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2									3
CO2	2	3	2									3
CO3	3	2	2									3
CO4	3	3	3									3
CO5	3	3	3									3

KTU SPECIAL

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation*

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Programming languages: principles and practices.	Kenneth C Louden	Cengage Learning	3/e, 2011
2	Concepts of programming languages.	Sebesta R W.	Pearson	12/e, 2023
3	Programming languages: concepts and constructs.	Sethi R	Pearson	2/e, 2006

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Programming Languages: Principles and Paradigms	Allen Tucker, Robert Noonan	McGraw-Hill	2/e, 2017
2	Principles of programming languages.	Gilles Dowek.	Springer	1/e, 2009.
3	Principles of Programming Languages	Rajiv Chopra	Wiley	1/e, 2019

Video Links (NPTEL, SWAYAM…)	
No.	**Link ID**
1	https://archive.nptel.ac.in/courses/106/102/106102067/

KTU SPECIAL