SEMESTER S7

ADVANCED COMPUTER PROGRAMMING TECHNIQUES

Course Code	PEBMT746	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	PE

Course Objectives:

1. To introduce the various programming paradigms.
2. To understand the evolution of programming languages.
3. To understand the concepts of languages, functional languages, logical and scripting languages.
4. To introduce the principles and techniques involved in design and implementation of modern programming languages.
5. To introduce the notations to describe the syntax and semantics of programming languages.

KTU SPECIAL

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	Preliminary Concepts: Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, 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	9

2	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, iterative statements, unconditional branching, guarded commands.	**9**
3	Subprograms: Fundamentals of subprograms, design issues for subprograms, local referencing environments, parameter passing methods, parameters that are subprograms, calling subprograms indirectly, overloaded subprograms, generic subprograms, design issues for functions, 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	**9**
4	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#. 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.	**9**

KTUSPECIAL

Course Assessment Method

(CIE: 40 marks , ESE: 60 marks)

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 =24marks)**	• 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 sub divisions. **(4x9 = 36 marks)**	**60**

KTU SPECIAL

Course Outcomes (COs)

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

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Understand to express syntax and semantics in formal notation	**K2**
CO2	Employ to apply suitable programming paradigm for the application.	**K3**
CO3	Design to program in different language paradigms and evaluate their relative benefits	**K6**
CO4	Understand the programming paradigms of modern programming languages.	**K2**
CO5	Understand the concepts of ADT and OOP	**K2**
CO6	Knowledge to compare the features of various programming languages	**K1**

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

CO-PO Mapping Table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	1					1		2
CO2	3	3	2	2	1					1		2
CO3	3	3	2	2	1					1		2
CO4	3	3	2	2	1					1		2
CO5	3	3	2	2	1					1		2
C06	3	3	2	2	1					1		2

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

KTU SPECIAL

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Programming Languages	A.B. Tucker, R.E. Noonan	TMH	
2	Programming Languages	K. C. Louden and K A Lambert	Cengage Learning	3rd edition
3	Programming Language Concepts	C Ghezzi and M Jazayeri	Wiley India	
4	Programming Languages	Ravi Sethi	Pearson	2nd Edition

Video Links (NPTEL, SWAYAM…)	
Module No.	**Link ID**
Module - II	https://nptel.ac.in/courses/122104019
Module - IV	https://nptel.ac.in/courses/106106210

KTU SPECIAL