17CS2206 - PRINCIPLES OF PROGRAMMING LANGUAGES

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Category:** |  Core | **Credits:** | 4 |
| **Course Type:** | Theory | **Lecture – Tutorial – Practical:** | 3-2-0 |
| **Prerequisite:** | A few basics regarding syntax and semantics of programming languages | **Sessional Evaluation:****Univ. Exam Evaluation:****Total Marks:** | 4060100 |
| **Objectives** | * To understand various basic Programming language preliminaries
* To describe a common syntax and semantics among languages
* To explore the basic features of Object Orientation and a few merits from functional programming languages
* To get better exposure regarding the comparative study of various languages
 |

|  |  |
| --- | --- |
| **Course Outcomes** | Upon successful completion of the course, the students will be able to: |
| CO1 | Identify the basic language evaluation criteria among multiple Programming Domains. |
| CO2 | Study various simple building blocks to construct different statements. |
| CO3 | Come across local referencing environments and parameter passing mechanisms. |
| CO4 | Learn the concept of abstraction and encapsulation properties.  |
| CO5 | Acquire the list of essential elements of object oriented programming languages.  |
| CO6 | Identify various Exception Handling mechanisms and exploring Functional programming features. |
| **Course Content** | UNIT – I**Preliminaries:** Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Tradeoffs, Implementation methods and programming environments.**Describing Syntax and Semantics:** Formal methods of describing Syntax, Attribute Grammars and Dynamic Semantics.UNIT – II**Data Types**: Primitive data types, Character string types, User-defined ordinal types, Arrays, Associative arrays, Record and Union types, Pointer and reference types. **Expression statements and Assignment statements**: Introduction, Arithmetic expressions, Overloaded operators, Type conversions, Relational and Boolean expressions, Short-Circuit evaluation, Assignment and Mixed mode statements.UNIT – III**Subprograms**: Fundamentals, Design issues, Local referencing environments, Parameter passing methods, Parameters that are subprogram names, Overloaded and Generic subprograms, Design issues for functions, User-defined Overloaded Operators, Co routines. **Implementing Subprograms:** General semantics of calls and returns, Implementing simple subprograms, Subprogram implementation with stack dynamic local variables.UNIT – IV**Abstract Data Types:** Concepts of abstraction, Data Abstraction, Design issues, Language Examples, Parameterized Abstract data types. **Encapsulation Constructs**: Introduction, Encapsulation Constructs, Naming Encapsulations.UNIT – V**OOP features**: Introduction, Object oriented programming concepts, Design Issues. **OOP Languages:** Support for OOP in Smalltalk, C++, Java, C#, Ada95, Ruby and Object model of JavaScript, Implementation of Object Oriented constructs.UNIT –VI**Exception Handling**: Introduction, Exception handling In Ada, C++ and Java.**Functional Programming Languages**: Overview, Fundamentals of Mathematical functions, introduction to LISP and Scheme. Applications, Comparison of functional and imperative languages. |
| **Text Books and References:** | **Text Books:**1. Sebesta RW, Concepts of programming languages, 7th edition, Pearson Education 2008

**Reference Books:**1. Louden KC, Programming Languages - Principles and Practice, 2nd edition, Cenage Learning 2003.
2. Tucker AB,Noonan RE, Programming Languages- Principle and Practices, 2nd Edition
3. Pratt TW, Zelkowitz MV and Gopal TV, Programming Languages - Design and Implementation, 4th Edition, Pearson Education 2006.
4. Ghezzi C. Jazayeri M, Programming Language Concepts, 3rd Edition, Wiley -India 1998.
 |
| **E-Resources** | 1. [**https://nptel.ac.in/courses**](https://nptel.ac.in/courses)
2. [**https://freevideolectures.com/university/iitm**](https://freevideolectures.com/university/iitm)
 |