|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 13CS3103 | - | PRINCIPLES OF PROGRAMMING LANGUAGES | | | | | | | | |
|  |  | |  | | | | |
| Hours / Week | : | 4 | |  | Sessional Marks | : | 40 |
| Credits | : | 4 | |  | End Examination Marks | : | 60 |

|  |
| --- |
| **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, Array types, Associative arrays, Record types, Union types, Pointer and reference types.  **Expression statements and Assignment statements**: Arithmetic expressions, Overloaded operators, Type conversions, Relational and Boolean expressions, Short-Circuit evaluation, Assignment statements, Mixed mode assignment. |
|  |
| **UNIT – III** |
| **Subprograms**: Fundamentals, Design issues, Local referencing environments, Parameter passing methods, Parameters that are subprogram names, Overloaded subprograms, Generic subprograms, Design issues for functions, User-defined Overloaded Operators, Co routines.  **Implementing Subprograms**: General Semantics of calls and returns, implementing simple subprograms, implementing subprograms with stack-dynamic local variables. |
|  |
| **UNIT – IV** |
| **Abstract Data Types & Encapsulation Constructs**: Concepts of abstraction, Data Abstraction, Design issues, Language Examples, Parameterized Abstract data types, Encapsulation Constructs, Naming Encapsulations.  **Support for OOP**: Object Oriented Programming, Design Issues for OOL, Support for OOP in Smalltalk, C++, Java, C#, Ada95, Ruby and Object model of JavaScript, Implementation of Object Oriented constructs. |
|  |
| **UNIT - V** |
| **Exception Handling**: Introduction to Exception Handling, Exception handling In Ada, C++, Java.  **Functional Programming Languages**: Mathematical functions Fundamentals of functional programming languages, introduction to LISP, Scheme. Applications of functional programming languages, Comparison of imperative languages. |
|  |
| TEXT BOOKS |
| 1. Sebesta RW, Concepts of programming languages, 8th 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 - Principles and Paradigms, 2nd edition, TMH 2007. 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. |