

**DEPARTMENT OF CSE & IT**

**LECTURE SCHEDULE**

# CLASS : II B.TECH - II SEMESTER REGULATION: R16

# BRANCH : IT

**SUBJECT** : Principles of Programming Languages

**ACADEMIC YEAR** : 2017 - 2018

**FACULTY** : Mrs. M.DEEPTHI

|  |  |  |
| --- | --- | --- |
| **UNIT No. & Name** | **Topic** | **No. Of Classes**  (Hours required) |
| **UNIT - I :**  **SYNTAX AND SEMANTICS** | **Lecture 1:** Why to study Programming languages  **Lecture 2,3,4,5:** Evolution of programming languages  **Lecture 6:** Describing syntax  **Lecture 7:** CFG,Parse Trees,Ambiguity  **Lecture 8:** EBNF,Attribute Grammer  **Lecture 9,10,11:** Describing semantics  **Lecture 12:** Linear and binary recursion,  **Lecture 13:** Lexical analysis  **Lecture 14:** parsing  **Lecture 15,16:** Top down and bottom - up parsing | 16 |
| **UNIT - II :**  **DATA, DATA TYPES, AND BASIC STATEMENTS** | **Lecture 17:** Names, variables  **Lecture 18:** binding, type checking, scope, scope rules, lifetime  **Lecture 19:** garbage collection, primitive data types, strings  **Lecture 20:** array types, associative arrays, record types, union types  **Lecture 21:**pointers and references  **Lecture 22:** Arithmetic expressions  **Lecture 23:** overloaded operators, type conversions  **Lecture 24:** relational and boolean expressions, assignment statements , mixed mode ssignments  **Lecture 25,26:** control structures – selection, iterations, branching, guarded Statements | 10 |
| **UNIT - III :**  **SUBPROGRAMS AND IMPLEMENTATIONS** | **Lecture 27:** Subprograms, design issues  **Lecture 28:** Local referencing, parameter passing,  **Lecture 29:** Overloaded methods, generic methods  **Lecture 30:** Design issues for functions  **Lecture 31,32:** Semantics of call and return, implementing simple Subprograms  **Lecture 33: S**tack and dynamic local variables  **Lecture 34,35: N**ested subprograms, blocks, dynamic scoping | 9 |
| **UNIT - IV :**  **OBJECT- ORIENTATION, CONCURRENCY, AND EVENT HANDLING** | **Lecture 36:** Object – orientation, design issues for OOP languages.  **Lecture 37,38:** Implementation of object, oriented constructs  **Lecture 39:** Concurrency  **Lecture 40:** Semaphores  **Lecture 41,42:** Monitors, message passing, threads  **Lecture 43:** Statement level concurrency  **Lecture 44,45:** Exception handling, event handling | 10 |
| **UNIT - V:**  **FUNCTIONAL PROGRAMMING LANGUAGES** | **Lecture 46,47:** Introduction to lambda calculus  **Lecture 48,49:** fundamentals of functional  programming languages  **Lecture 50,51:** Programming with Scheme,  **Lecture 52,53:** Programming with ML | 8 |
| **UNIT - VI :**  **LOGIC PROGRAMMING LANGUAGES** | **Lecture 54,55:** Introduction to logic and logic programming  **Lecture 56,57,58:** Programming with Prolog  **Lecture 59,60:** Multi - paradigm languages | 7 |
| **Total number of classes required: 60** | | |

**TEXT BOOKS:**

1. Robert W. Sebesta, “Concepts of Programming Languages”, Tenth Edition, Addison Wesley, 2012.

2. Programming Langugaes, Principles & Paradigms, 2ed, Allen B Tucker, Robert E Noonan, TMH

**Signature of the Faculty Signature of the HOD**