**USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY**

***Department of Computer Science and Engineering***

**LESSON PLAN:: PPL**

|  |  |
| --- | --- |
| **Academic Year** : 2016-17 | **Sem**  : I |
| **Course**: Principles of Programming Languages(PPL) R13 | |
| **Class**: III B.TECH. | **Section** : CSE - B |
| **Date of commencement of Class work** :13-6-2016 | **Date of end of Class work** :8-10-2016 |
| **Prepared By**: G. Baleswari Assistant Professor | **Approved By**: HOD |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **TOPIC** | **MODE OF DELIVERY** | | **DATE** |
| **UNIT-I** | | | | |
| **Introduction to PPL, SYNTAX AND SEMANTICS** | | | | |
| 1 | Introduction to PPL | Lecture interspersed with discussion | 13-6-2016 | |
| 2 | Syntax and Semantics | 14-6-2016 | |
| 3 | Evolution of programming languages | 15-6-2016 | |
| 4 | describing syntax, | 16-6-2016 | |
| 5 | Context free grammars | 17-6-16,20 -6-16 | |
| 6 | Attribute grammars, | 20-6-16, 21-6-16 | |
| 7 | describing semantics | 22-6-16, 23-6-16 | |
| 8 | Lexical analysis | 24-6-16, 27-6-16 | |
| 9 | Parsing | 27-6-16, 28-6-16 | |
| 10 | recursive - decent bottom up parsing | 29-6-16,30-6-16 | |
| 11 | revision | 1-7-2016 | |
| 12 | Tutorial/Unit test | 1-7-2016 | |
| **UNIT-II** | | | | |
| **Data, Data Types, and Basic Statements** | | | | |
| 13 | Names, variables, binding | Lecture interspersed with discussion | | 4-7-2016 |
| 14 | type checking, | 5-7-2016 |
| 15 | scope, scope rules | 7-7-2016 |
| 16 | lifetime and garbage collection, | 8-7-2016 |
| 17 | primitive data types | 8-7-16,11-7-16 |
| 18 | strings | 11-7-2016 |
| 19 | array types, | 12-7-2016 |
| 20 | associative arrays | 13-7-2016 |
| 21 | record types, union types, | 14-7-2016 |
| 22 | pointers and references, | 15-7-2016 |
| 23 | Arithmetic expressions | 18-7-2016 |
| 24 | overloaded operators, type conversions | 18-7-2016 |
| 25 | relational and Boolean expressions | 19-7-2016 |
| 26 | assignment statements , | 20-7-2016 |
| 27 | mixed mode assignments | 21-7-2016 |
| 28 | control structures – selection, iterations, | 22-7-2016 |
| 29 | branching, guarded Statements | 25-7-2016 |
| 30 | Revision ,Tutorial/Unit test | 25-7-2016 |
| **UNIT-III** | | | | |
| **Subprograms and Implementations** | | | | |
| 31 | Subprograms , design issues, | Lecture interspersed with discussion | | 26-7-2016 |
| 32 | local referencing, | 28-7-2016 |
| 33 | parameter passing | 28-7-2016 |
| 34 | overloaded methods, generic methods | 29-7-2016 |
| 35 | design issues for functions, | 30-7-2016 |
| 36 | semantics of call and return, | 1-8-2016 |
| 37 | implementing simple subprograms | 2-8-2016 |
| 38 | stack and dynamic local variables | 3-8-2016 |
| 39 | nested subprograms, | 4-8-2016 |
| 40 | blocks, dynamic scoping | 5-8-2016 |
| 41 | Revision Tutorial/Unit test | 5-8-2016 |
| **UNIT-IV** | | | | |
| **Object- Orientation, Concurrency, and Event Handling** | | | | |
| 42 | Object – orientation | Lecture interspersed with discussion | | 17-8-2016 |
| 43 | design issues for OOP languages, | 19-8-2016 |
| 44 | Problems related to design | 22-8-2016 |
| 45 | implementation of object-oriented constructs | 22-8-2016 |
| 46 | concurrency | 23-8-2016 |
| 47 | semaphores | 23-8-2016 |
| 48 | Monitors, | 26-8-2016 |
| 49 | message passing, | 27-8-2016 |
| 50 | threads, | 29-8-2016 |
| 51 | statement level concurrency | 30-8-2016 |
| 52 | exception handling, | 31-8-2016 |
| 53 | event handling | 1-9-2016 |
| 54 | Revision | 2-9-2016 |
| 55 | Tutorial/Unit test | 2-9-2016 |
| **UNIT-V** | | | | |
| **Functional Programming Languages** | | | | |
| 56 | Introduction to lambda calculus | Lecture interspersed with discussion | | 6-9-2016 |
| 57 | More examples on lambda calculus | 7-9-16, 8-9-16 |
| 58 | Introduction to functional programming languages | 9-9-2016 |
| 59 | fundamentals of functional programming languages | 12-9-2016 |
| 60 | Compare functional programming languages | 13-9-2016 |
| 61 | Introduction to Scheme | 14-9-2016 |
| 62 | Programming with Scheme | 15-9-16, 16-9-16 |
| 63 | Introduction to ML, | 16-9-2016 |
| 64 | Programming with ML, | 19-9-16, 20-9-16 |
| 65 | Revision | 21-9-2016 |
| 66 | Tutorial/Unit test | 22-9-2016 |
| **UNIT-VI** | | | | |
| **Logic Programming Languages** | | | | |
| 67 | Introduction to logic | Lecture interspersed with discussion | | 23-9-2016 |
| 68 | Introduction to logic programming | 26-9-2016 |
| 69 | Introduction to Prolog | 26-9-16, 27-9-16 |
| 70 | Program constructs of Prolog | 28-9-2016 |
| 71 | Programming with Prolog | 29-9-16, 30-9-16 |
| 72 | Introduction to multi - paradigm languages | 3-10-2016 |
| 73 | Some constructs of these languages | 3-10-16, 4-10-16 |
| 74 | Some examples of multi-paradigm languages | 5-10-2016 |
| 75 | Tutorial/Unit test | 6-10-2016 |
| 76 | Revision | 7-10-2016 |

**\*I-MID 8/8/2016 TO 13/08/2016 \*\* II-MID 10/10/2016 TO 15/10/2016**

**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

**REFERENCES:**

1. R. Kent Dybvig, “The Scheme programming language”, Fourth Edition, MIT Press, 2009.

2. Jeffrey D. Ullman, “Elements of ML programming”, Second Edition, Prentice Hall, 1998.

3. Richard A. O'Keefe, “The craft of Prolog”, MIT Press, 2009.

4. W. F. Clocksin and C. S. Mellish, “Programming in Prolog: Using the ISO Standard”, Fifth Edition, Springer, 2003

**List the Course Outcomes(Cos):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sub code | Sub Name | COs | Expected level of attainment  On 5 scale |
|  | Principles of Programming Languages | 1.Describe syntax and semantics of programming languages  2.Explain data, data types, and basic statements of programming languages  3.Design and implement subprogram constructs, Apply object - oriented, concurrency, and event handling programming constructs  4. Develop programs in Scheme, ML, and Prolog.  5.Understand and adopt new programming languages. | 3.5  3.5  3.5  3.5  3.5 |

**SIGNATURE OF FACULTY SIGNATURE OF HOD**