



**Department of Computer Science and Engineering
National Institute of Technology Tiruchirappalli**

COURSE PLAN

| 1. Course Outline | | | |
|-----------------------------------|-------------------------------------|-----------------------|-------------------------|
| Course Title | Principles of Programming Languages | | |
| Course Code | CSPC23 | | |
| Department | CSE | No. of Credits | 3 |
| Pre-requisites Course Code | NIL | Faculty Name | Mrs.T.Siron Anita Susan |
| E-mail | siron@nitt.edu | MobileNo. | 9488269859 |
| Course Type | Core Course | | |

| 2.Course Overview |
|---|
| Principles of Programming Languages mainly describes about the syntax, semantics and implementation of various programming language paradigms. |
| 3. Course Objectives |
| <ul style="list-style-type: none"><input type="checkbox"/> To understand and describe syntax and semantics of programming languages<input type="checkbox"/> To understand data, data types, and basic statements<input type="checkbox"/> To understand call-return architecture and ways of implementing them<input type="checkbox"/> To understand object-orientation, concurrency, and event handling in programming languages<input type="checkbox"/> To develop programs in non-procedural programming paradigms |
| 4. Course Outcomes (CO) |
| <ul style="list-style-type: none"><input type="checkbox"/> Describe syntax and semantics of programming languages<input type="checkbox"/> Explain data, data types, and basic statements of programming languages<input type="checkbox"/> Design and implement subprogram constructs<input type="checkbox"/> Apply object-oriented, concurrency, and event handling programming constructs<input type="checkbox"/> Develop programs in Scheme, ML, and Prolog<input type="checkbox"/> Understand and adopt new programming languages |

| 5. Course Outcome (CO) | Aligned Programme Outcome (PO) | | | | | | | |
|---|--------------------------------|------|------|------|------|------|------|------|
| | PO-1 | PO-2 | PO-3 | PO-4 | PO-5 | PO-6 | PO-7 | PO-8 |
| Describe syntax and semantics of programming | S | B | M | M | B | M | B | M |
| Explain data, data types, and basic statements of programming languages | S | B | M | B | M | B | B | M |
| Design and implement subprogram constructs | M | B | S | S | M | M | M | B |
| Apply object-oriented, concurrency, and event handling programming constructs | S | M | M | M | S | B | M | M |
| Develop programs in Scheme, ML, and Prolog | S | M | M | M | B | M | M | B |
| Understand and adopt new programming languages | M | M | B | B | S | B | M | B |

S = 0.6

M = 0.4

B = 0.0

| 6. Course Teaching and Learning Activities | | | |
|--|-------|--|----------------------|
| Weeks | Hours | Title | Mode of delivery |
| 1. | 1 | Evolution of programming languages – describing syntax & semantics | Chalk and board |
| | 2 | Lexical analysis –Parsing –recursive-decent –bottom up parsing | Chalk and board |
| | 3 | Primitive data types –strings | Chalk and board |
| 2. | 4 | Array types associative arrays –record types | Chalk and board |
| | 5 | Union types –Pointers and references | Chalk and board |
| | 6 | Arithmetic expressions –relational and Boolean expressions | Chalk and board |
| 3. | 7 | Assignment statements | Chalk and board |
| | 8 | Mixed-mode assignments | Chalk and board |
| | 9 | Control structures –Selection –Iterations | Chalk and board |
| 4. | 10 | Branching –guarded statements | Chalk and board |
| | 11 | Programming Assignments | PPT |
| | 12 | Subprograms –Design issues | Chalk and board |
| 5. | 13 | Local referencing –Parameter passing | Chalk and board, PPT |
| | 14 | Overloaded methods –generic methods | Chalk and board, PPT |
| | 15 | Design issues for functions | Chalk and board |
| 6. | 16 | Semantics of call and return –implementing simple subprograms | Chalk and board, PPT |
| | 17 | Dynamic local variables –Nested subprograms | Chalk and board, PPT |

| | | | |
|-----|----|---|----------------------|
| | 18 | Blocks –Dynamic scoping | Chalk and board |
| 7. | 19 | Programming Assignments | PPT |
| | 20 | Object-orientation –design issues for OOP languages | Chalk and board |
| | 21 | Implementation of object-oriented constructs –Concurrency | Chalk and board, PPT |
| 8. | 22 | Semaphores | Chalk and board, PPT |
| | 23 | Monitors | Chalk and board |
| | 24 | Message passing | Chalk and board |
| 9. | 25 | Threads –statement level concurrency | Chalk and board |
| | 26 | Exception handling –even handling | Chalk and board |
| | 27 | Programming Assignments in C++ | Chalk and board, PPT |
| 10. | 28 | Introduction to lambda calculus | Chalk and board |
| | 29 | Fundamentals of Functional programming languages | Chalk and board |
| | 30 | Programming with Scheme –Introduction to LISP | Chalk and board |
| 11. | 31 | Lists - Storage allocation for lists | Chalk and board |
| | 32 | Some useful functions - Error handling | Chalk and board |
| | 33 | Programming Assignments in LISP | Chalk and board,PPT |
| 12. | 34 | Introduction to logic and logic programming- | Chalk and board |
| | 35 | Computing with relations | Chalk and board |
| | 36 | Programming with Prolog - Intoduction | Chalk and board, PPT |
| 13. | 37 | Data structures in Prolog | Chalk and board, PPT |
| | 38 | Programming techniques - Control in Prolog | Chalk and board |
| | 39 | Cuts .– Multi-paradigm languages | Chalk and board |
| 14. | 40 | Programming Assignments in PROLOG | Chalk and board, PPT |

7. Course Assessment Methods

| Sl. No. | Mode of Assessment | Week/Date | Duration | Marks |
|---------|--------------------|--|----------|-------|
| 1 | Cycle Test – 1 | 5 th week | 1 hour | 20 |
| 2 | Cycle Test – 2 | 11 th week | 1 hour | 20 |
| 3 | Assignment | 4 th , 10 th weeks | – | 10 |
| 4 | End Semester Exam | November last week | 3 hours | 50 |
| Total | | | | 100 |

8. Essential Readings

Text Books

- Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
- Michael L. Scott, "Programming Language Pragmatics", Third Edition, Morgan Kaufmann, 2009.
- R. Kent Dybvig, "The Scheme programming language", Fourth Edition, MIT Press, 2009.
- Jeffrey D. Ullman, "Elements of ML programming", Second Edition, Prentice Hall, 1998.
- Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009.
- W. F. Clocksin and C. S. Mellish, "Programming in Prolog: Using the ISO Standard", Fifth Edition, Springer, 2003.

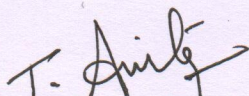
Course Exist Survey

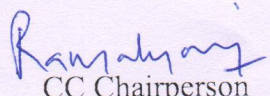
Student feedback form will be collected at the end of the course through MIS

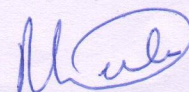
Course Policy

Attendance- Students having 75% to 100% attendance are eligible for writing the End semester Examination. Students having attendance between 65% & 75% with valid reasons can write the end semester exam after attending extra classes. Students having less than 65% have to redo the course. Student should not absent for the assessment. If the reason for absence is genuine, the student can reappear for reassessment.

For Senate's Consideration


Course Faculty


CC Chairperson
7/7/12


HOD