

BITS PILANI, DUBAI CAMPUS
ACADEMIC – UNDERGRADUATE STUDIES DIVISION
First Semester 2023 – 2024
Course Handout (Part – II)

Date: 28.08.2023

In addition to Part I (General Handout for all courses appended to the Time Table) this portion further specific details regarding the course.

Course No : CS F301 (2 0 2)
Course Title : Principles of Programming Languages
Course Instructors : Dr B. Vijayakumar, Dr Pranav M Pawar
Instructor-in-charge : Dr B. Vijayakumar

Scope and Objectives:

The course covers feature of programming languages and introduces the main programming paradigms. It covers, in detail, the semantics of the features of programming languages –Control Abstraction, Data Types and Data Abstraction, Scope and Parameter passing and Concurrency related features. It covers various aspects of runtime environments like global and local data, code, function call stacks, dynamically allocated data, runtime features for exceptions and threads. Introduction to programming paradigms. Functional paradigm – formal elements of lambda calculus, introduction to syntax of common functional programming languages and programming exercises that explore the functional paradigm. Logic programming paradigm - formal elements of logic programming and programming tasks that explore the logic paradigm. Scripting as a paradigm. Domain specific languages.

Course Pre/Co- requisite (if any) & Catalogue / Bulletin Description: *Given in the Bulletin 2020 – 2021 – Data Structures and Algorithms.*

Text Books [TB]:

- i. Ravi Sethi, "Programming Languages: Concepts and Constructs" 2nd Edition by Addison Wesley, 2006 (Reprint 2010).

Reference Books [RB]:

- i. David A Watt, William Findlay, "Programming Language Design Concepts", John Wiley & Sons, Ltd., 2004.
- ii. Aho, Lam, Sethi and Ullman, "Compilers Principles, Techniques, and Tools". Pearson Education Edition. 2004.
- iii. Ellis Horowitz, "Fundamentals of Programming Languages", Galgotia Pub. Ltd., 2004.
- iv. Robert W. Sebesta, "Concepts of Programming Languages", 9th Edition by The Benjamin/Cummings Publishing Company, Inc., 2009.
- v. Programming Languages, *Principles and Practice 3rd Edition*, by Kenneth C. Louden & Kenneth A. Lambert, Cengage Learning 2012. (online)
- vi. Introduction to Programming Languages, 1st Edition, Arvind Kumar Bansal, Chapman and Hall/CRC, Published December 14, 2013. (online)
- vii. Michael L.Scott, Programming Language Pragmatics 3rd edition.
- viii. Paul Barry. 2016. Head First Python, 2nd. ed.

Research topics:

1. Research issues in programming languages
2. Programming Languages for High Performance Computing

Course Plan / Schedule Sec 1: Tue 3, Wed 2 Sec 2: Tue 3, Wed 2 Sec 3: Mon 2, Tue 5

S. No.	Topic	Learning Objectives	No of Lectures	Reading
1	Introduction and Motivation	Historic background, Criteria for a good programming language design	1	T (i): Ch 1, R (iii): Ch. 1

S. No.	Topic	Learning Objectives	No of Lectures	Reading
2	Language Paradigms	Imperative, Declarative, Features of a Programming Language, compilers	1	T (i): Ch 1, Class Notes
3	Language Description	Syntactic Structure	2	T(i): Ch.2
4	Statements: Structured Programming	Control Abstraction, Loops, Jumps	1	T(i): Ch 3
5	Types, Data Layout Models	Data Representation, Primitive and Structured Data Types, ADTs, Type checking vs. Type Inferencing, Type Equivalence and Subtyping, Data Layout Models	3	T(i): Ch 4
6	Basic Runtime Environments and Procedure Activations	Code vs. Data, Global vs. Local Data, Functions and Call Stacks, dynamically allocated data and heaps; Introduction to procedures, Recursion, Parameter Passing Methods, Call-by-Value, Call-by-Reference. Scope rules for names, static and dynamic scope rules, nested scope, Activation Records, Lexical Scope, Dangling Pointers, Tail Recursion	3	Class Notes, R (ii): Ch 7 T(i): Ch 5
7	Object Oriented Abstraction	Object Oriented Programming Paradigm, and features, Class hierarchy, Inheritance, Information hiding, Single and Multiple Inheritance, Polymorphism	3	T(i): Ch 7
8	Scripting as a paradigm, emerging trends	Problem domains, Scripting in WWW, Python: Basics, data types, loops, list, tuple, dictionary, functions. Classes, Objects, Inheritance.	3	Class notes, R(viii)
9	Functional Programming	Functional Programming with Lists (using PICO LISP): Characteristics, atoms, lists, procedures, primitive functions, expressions, List Operations, Box and Arrow Notation, record structure, procedure abstraction, de, setq, cond, recurrence relations in PICO LISP, read, while, if, print	2	T(i): Ch 8 T(i): Ch 9,10
11	Logic Programming	Logic Programming, The Structure of Prolog programs, Types/Terms, Type Predicates, Expressions, Unification, Functions, Recurrence Relation in Prolog, Rules, Boolean Predicates, Logical Operators, Functions-Computing with Relations, Database implementation in Prolog.	2	T(i): Ch 11
12	Concurrent Programming	Threads, IPCS overview, Shared Memory, Critical Section problem, Synchronization Features (semaphores, monitors, locks), Dining Philosophers Problem and its solution using monitors, Shared Memory programming	3	T(i): Ch 12 R (iv)
13	Complex Run Time Environments	Exception handling and non-local jumps, Threads	2	Class Notes Ch 12:R (iv)
14	Advanced Topics	Advancements in Programming Paradigm, Domain specific programming, research topics, Open MPI parallel programming library	2	Class notes
Total No of Classes Planned			28	

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students should be able to:

- **CLO1** To understand the basic theory of programming languages, language features and paradigms of different programming languages.
- **CLO2** To be able to describe syntax, semantics, names, scopes, bindings, control flow, data types and execution of programming languages.
- **CLO3** To understand the concepts of OO languages, functional languages, logical and scripting languages.
- **CLO4** To determine appropriate programming languages for given applications and apply it effectively.
- **CLO5** To introduce the concepts of concurrent programming, complex runtime environment and advanced topics.

Evaluation Scheme:

EC No	Components	Duration	Weightage %	Date & Time	Venue
1	Quiz 1 (Closed Book)	20 minutes	10%	05.10.23 Th 2	TBA
2	Midsem Exam (Open Book)	90 minutes	30%	31.10.23 AN	
3	Mini Project (Open Book)	-	20%	26.11.23	
4	Comprehensive Exam (Closed Book)	3 hours	40%	04.01.24 AN	

Mini Project: Mini project may be chosen by students on either some or all of the above-mentioned topics. Case studies, interpretation of data and then analysis, will form a part of all evaluation components. Mini-Project evaluation includes coding, seminar, presentation and viva. The assignments will involve use of C/C++/LISP/PROLOG/Python, research topics, Open Source Tools, and so on. Details will be intimated through a separate notification or announced in the class.

Tutorials: Tutorial sheets will be distributed in classes. This will help you to enhance your abilities in problem solving. Assistance will be given in solving them during the classes. No make-up will be provided for tutorials.

Mapping of CLOs, PLOs, and CECs

CLOs	PLOs	Evaluation Components (ECs)			
		EC1	EC2	EC3	EC4
CLO1	1,2	✓	✓		✓
CLO2	2,3,4	✓	✓		✓
CLO3	2,3,4,5,6		✓		✓
CLO4	5,6,8			✓	✓
CLO5	1,2,3				✓

* Please refer the [link](#) for the PLOs of the B.E. Computer Science programme

Mid-sem Grading:

Mid-sem grading will be displayed after two evaluation components or earlier when-ever about 40% of evaluation components are completed.

Note: A student will be likely to get “NC”, if he / she doesn't appear / appear for the sake of appearing for the evaluation components / scoring zero in pre-compre total.

Makeup and Attendance policies:

Make-ups: are not given as a routine. It is solely dependent upon the genuineness of the circumstances under

which a student fails to appear in a scheduled evaluation component. In such circumstances, prior permission should be obtained from the Instructor-in-Charge (I/C). Students with less than 60% of attendance will not be allowed to avail the make-ups. The decision of the I/C in the above matter will be final.

Attendance: Every student is expected to be responsible for regularity of his/her attendance in class rooms and laboratories, to appear in scheduled tests and examinations and fulfill all other tasks assigned to him/her in every course. A student should have a minimum of 60% of attendance in a course to be eligible to appear for the Comprehensive Examination in that course. For the students under the purview of Academic Counseling Board (ACB), the Board shall prescribe the minimum attendance requirement on a case-to-case basis. Attendance in the course will be a deciding factor in judging the seriousness of a student which may be directly / indirectly related to grading.

General timings for consultation:

Students can contact the course instructors in his/her chamber
273 (BVK), Tue 8 and 9th hour 1.55pm to 3.40pm.
238 (PMP), Tue 9th hour 2.50pm-03.40 pm.

General instructions:

Students should come prepared for classes and carry the text book(s) or material(s) as prescribed by the Course Faculty to the class.

Notices:

All Notices regarding this course will be placed on Computer Science Department noticeboard.

**Instructor-in-charge
CS F301**

Instructor's Contact Details:

- Dr. B. Vijayakumar, Professor, CS, WING B, Room No. 236
Contact No: +9714-4200699 Ext. no: 342 e-mail: vijay@dubai.bits-pilani.ac.in
- Dr Pranav M Pawar, Asst. Professor, CS, WING B, Room No. 238
Contact No: +9714-4200699 Ext. no: 304 e-mail: pranav@dubai.bits-pilani.ac.in