



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

FIRST SEMESTER 2020-21
COURSE HANDOUT

Date: 17.08.2020

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

Course No	: CS F301
Course Title	: Principles of Programming Languages
Instructor-in-Charge	: Vandana Agarwal (email: vandana@pilani.bits-pilani.ac.in)
Instructor(s)	: Amit Dua (email: amit.dua@pilani.bits-pilani.ac.in)
Tutorial Instructors	: Amit Dua, Vandana Agarwal, Sameen Ahmad

1. Course Description: The course covers features 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, inheritance, polymorphism 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. Different programming paradigms, namely imperative, object oriented, logic and functional programming will be discussed in detail. Students will be introduced to the significance of a construct and how different languages implement that. The Lambda calculus will follow an introduction to functional programming languages. Logic programming paradigm - formal elements of logic programming and programming tasks that explore the logic paradigm. Logic in automated theorem proving will be explained using appropriate examples.

2. Scope and Objective of the Course:

- Develop familiarity with the different paradigms of languages (imperative, functional, object oriented and logic) in programming and understanding the features of programming languages like abstraction, data types, etc.
- Understand the aspects of run-time environments like global and local data, function call stacks and static and dynamic allocation of data.
- Introduction to the syntax and semantics of common functional programming languages and exposure to programming exercises in this domain.
- Students will be encouraged to learn the design aspects of languages belonging to different paradigms rather than just learning a new language for coding purpose. However, specific programming languages will be expected to be learnt by the students for working on assignment problems.
- Understand the basis of logic programming and basic elements of prolog.
- Understand synchronization features in concurrent programming.

3. Text Books:

[T1]. **Ravi Sethi**, "Programming Languages: Concepts and Constructs" 2nd Edition by Addison Wesley.

4. Reference Books:

[R1]. **Robert W. Sebesta**, "Concepts of Programming Languages", 10th Edition by Pearson Publishers.



[R2]. Benjamin C. Pierce, “Types and Programming Languages”, The MIT Press

[R3]. Aho, Lam, Sethi and Ullman, "Compilers Principles, Techniques, and Tools". Pearson Education. Low Price Edition. 2004.

5. Course Plan:

Module No. (Instructor's name)	Lecture Session	Reference	Learning outcomes
Introduction (Amit Dua)	L1-L2: Introduction and Motivation, Language Paradigms, Imperative vs. Declarative Styles of Programming; Programming Languages that support these styles, Compilers. Features of a Programming Language.	Chapter 1 (T1) Chapter 1, 2 (R1)	Identifying the language paradigms and language evaluation criteria
Introduction (Amit Dua)	L3-L4: Language Description, Syntactic Structure	Chapter 2(T1) Chapter 3, 4 (R1)	Understanding the formal methods of describing the syntax, resolving ambiguity in grammars and converting BNF to EBNF form and vice-versa
Imperative Languages Paradigm (Vandana)	L5-L6: Control Abstraction, Structured programming, Loops and jumps	Chapter 3 (T1) Chapter 8 (R1)	Understanding the need for Structured Programming and Syntax Directed Control Flow, programming with invariants
Imperative Languages Paradigm (Vandana)	L7-L8: Data Representation, Primitive and Structured Data types, ADTs; Type checking vs. Type Inferencing, Type Equivalence and Subtyping.	Chapter 4 (T1) Chapter 6 (R1)	Identifying the Primitive Data Types, User defined ordinal types and constructed types. Understanding the concepts of type checking
Imperative Languages Paradigm	L9-L10: Primitive Data, Structured Data (Arrays/Lists, Records/Structures,	Chapter 4(T1) Chapter 6 (R1)	Understanding the Variables and attributes of a variable, binding times, categories of



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

(Vandana)	Variants/Unions), Objects (Simple objects and classes, Inheritance Models – Single vs. Multiple, Interfaces and abstract/virtual classes).		variables, Static and Dynamic scoping
Imperative Languages Paradigm (Vandana)	L11-L14: Introduction to procedures, Recursion, Parameter Passing Methods, Call-by-Value, Call-by-Reference, Call by Value Result. Scope rules for names, static and dynamic scope rules, nested scope, Activation Records, Lexical Scope, Dangling Pointers, garbage collection.	Chapter 5 (T1) Chapter 5, 9, 10 (R1)	Understanding the run time environment, semantics of calls and returns, implementing subprograms, implementing static scoping and dynamic scoping
Functional Programming Paradigm (Amit Dua)	L15-L16: Introduction and basic elements of Functional Programming, Lists and associated Operations, Function Declaration, Predicate Functions, Tail recursive functions, Higher Order Functions, Polymorphism	Chapter , 9, 10 (T1) Chapter 16 (R1)	Understanding the basics of functional programming and benefits of functional programming.
Functional Programming Paradigm (Amit Dua)	L17-L18: Data Types and Lambda Calculus	Chapter 14 (T1)	Understanding lambda calculus, and using type systems as a way to reason about programs
Semantics and Verification (Amit Dua)	L19-20: abstract interpretation (verification of imperative programs, following the operational semantic rules for partial correctness)	To be announced	Understanding basics of semantics and verification of programs
Object Oriented Programming paradigm (Amit Dua)	L21-L22: Object Oriented Programming Paradigm and features, Class hierarchy, Inheritance, Information hiding.	Chapter 7 (T1) Chapter 12 (R1)	Understanding the OOP concepts and design issues, implementing dynamic binding and inheritance in OOP languages



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

Logic Programming Paradigm (Vandana)	L23-L24: Design aspects of Logic programming languages	Chapter 11(T1) Chapter 15 (R1)	Basic understanding of logic programming constructs
Concurrency and Distribution (Vandana)	L25-L26: Design aspects of concurrent programming	Chapter 12 (T1) Chapter 13 (R1)	Implementing cooperation and competition synchronization in concurrent programming.

6. Evaluation Scheme:

Component	Duration	Weightage (%)	Date & Time (Exact details to be uploaded on Nalanda)	Nature of component (Close Book/ Open Book)
Class tests (During classes, On Nalanda)	10 minutes (each)	10% (1% for each test)	During lecture or tutorial classes (Best 10 to be considered for final grading)	Open book
Quizzes (Take home, on Nalanda)	30 minutes (each)	10% (2% for each quiz)	Regular quiz (Best 5 to be considered for final grading)	Open book
Assignments I and II (take home, in groups)	10 days	20% (10% for each assignment)	Tentative start dates for assignments: September 25, 2020 (Ist) October 25, 2020 (IInd)	Open book
Test 1	30 minutes	10%	September	Open book
Test 2	30 minutes	10%	October	Open book
Test 3	30 minutes	10%	November	Open book
Comprehensive Examination	120 minutes	30%	As per AUGSD guidelines	Closed or Open book (based on mode of examination)

Class Tests: The students are expected to attend lectures and tutorials regularly. During the class, preferably towards the end of it, the students will be expected to answer about 5-8 questions based on the topic discussed. These tests will be conducted during any of the lectures or tutorial classes. Each test will be of maximum 10 minutes duration. Students are expected to ensure good network speed at their end as per the AUGSD guidelines. No makeup request will be entertained for the class tests and the best ten tests out of twelve or more tests will be considered for grading. Also, the class test time will not be extended for any reason.



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

Quizzes: There will be a quiz in a week or ten days at regular intervals. The quiz questions will be made available on Nalanda. Students are advised to revise the concepts taught in the class regularly by going through the prescribed text and reference books. It is of utmost importance that the quizzes be attempted with proper preparation. A fixed window of about 90 minutes will be made available for students to attempt the quiz from anywhere. However, once the quiz is started, a student will have to complete that in the prescribed time of 30 minutes. The quiz can be started anytime during the given time window, but will end after the defined time limit. Students are also advised to avoid the last minute upload due to possible network congestion. Students will be graded for the best five of the all quizzes uploaded. No makeup request will be entertained if a student misses the quiz. Also, the quiz time will not be extended for any reason.

Assignments I and II: The students will be required to write programs in languages such as OCaml, SCALA, C, JAVA etc for solving the given problems. The students will be asked to form groups of three or four students and do the assignments in teams. The assignment problem statement will be uploaded on Nalanda and teams of students will be expected to solve the problem by way of coordinating among themselves using the appropriate technology such as Google meet or Microsoft teams. Students are advised to find compatible partners to work with and register their names when asked for.

Tests I, II and III: These tests will also be online and are intended to test the learned knowledge of a student about the design aspects of the taught programming languages paradigms. It will comprise of variety of short answer type questions, which will be multiple choice, fill in the blanks, matching etc. or long answers type explanation based questions.

7. Chamber Consultation Hour: To be announced

8. Notices: <https://nalanda.bits-pilani.ac.in/>

9. Make-up Policy:

- **Permission of the Instructor-in-Charge is required** to take a make-up.
- **Make-up applications must be given to the Instructor-in-charge personally.**
- ***A make-up test shall be granted only in genuine cases where - in the Instructor's judgment - the student would be physically unable to appear for the test.***
- Requests for make-up for the comprehensive examination – under any circumstances – can only be made to Dean, Instruction Division.

10. Note (if any):

Instructor-in-charge
Course No.