Birla Institute of Technology & Science, Pilani
Hyderabad Campus

INSTRUCTION DIVISION
FIRST SEMESTER 2016-2017
Course Handout Part II

01-08-2016

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/IS F301*
Course Title	:	*Principles of Programming Languages*
Instructor-in-Charge	:	*Dr. Aruna Malapati (arunam@bits-hyderabad.ac.in)*

Scope and Objective of the Course:
Programming languages are the medium through which we describe computations. More specifically, we use the model provided by a programming language to discuss concepts, formulate algorithms, and reason about problem solutions. Programming languages define models tailored to thinking about and solving problems in intended application areas. This course covers the design and implementation of modern general-purpose programming languages.
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. Assignments and case study focus on a variety of programming languages to facilitate exposure to language design and related implementation issues.

Textbooks:
1. T1. Robert Sebesta , Concepts of programming languages 10th Edition

Reference books
1. R1 Ravi Sethi, Programming Languages - Concepts and Constructs Pearson Education. Low Price Edition.
2. R2 Michael L.Scott, Programming Language Pragmatics 3rd edition
3. R3 David A. Watt, Programming Language Design Concepts- John Wiley & Sons.
4. R4 Aho, Lam, Sethi and Ullman, "Compilers Principles, Techniques, and Tools". Pearson Education. Low Price Edition. 2004

Course Plan:

Lecture No.	Learning objectives	Topics to be covered	Chapter in the Text Book
1	• To understand the reason to study this course	Introduction and Motivation	
2	• To learn the key characteristics of programming paradigms and express algorithms in: Object-oriented languages, Imperative languages, Functional languages, Logic programming languages and Parallel or concurrent languages • Evaluate the features of a programming language.	Language Paradigms, Imperative vs. Declarative Styles of Programming; Programming Languages that support these styles, Compilers. Features of a Programming Language, compilers	Chapter 2 (T1) Chapter 1 (R1)
3	• To learn to read and write formal descriptions of programming language syntax.	Language Description: Syntactic Structure,	Chapter 3(T1) Chapter 2(R1)
5-7	• Understand concept of a data type and the characteristics of the common primitive data types. • Ability to check the equivalence of the data type and sub types using name and structural equivalence	Types: Data Representation, Primitive and Structured Data types, ADTs; Type checking vs. Type Inferencing, Type Equivalence and Subtyping.	Chapter 6 (T1) Chapter 4 (R1)
8-10	• Analyze and evaluate the design choices to implement derived data types for programming languages	Data Layout models: Primitive Data, Structured Data (Arrays/Lists, Records/Structures, Variants/Unions), Objects (Simple objects and classes, Inheritance Models – Single vs. Multiple, Interfaces and abstract/virtual classes).	Chapter 6 (T1)
11	• Analyze and evaluate control abstractions of programming languages • Select and apply appropriate expressions and control structures for a given programming task	Control Abstraction: Structured programming, Loops and jumps	Chapter 8 (T1) Chapter 3 (R1)
12-13	• Understand the importance how the of the run time memory model changes with respect to the features of a programming language.	Basic Runtime Environments: Code vs. Data, Global vs. Local Data, Functions and Call Stacks, Dynamically allocated data and heaps;	Chapter 9 & 10 (T4) Chapter 5 (R1)
14-16	• Understand, define and find the impact of different parameter passing mechanism, Scope and Binding	Procedures: Introduction to procedures, Recursion, Parameter Passing Methods, Call-by-Value, Call-by-Reference, Call by Value Result. Scope rules for names, static and dynamic	Chapter 9 & 10 (T4) Chapter 5 (R1)

		scope rules, nested scope, Activation Records, Lexical Scope, Dangling Pointers, Tail Recursion Elimination	
17-18		Object Oriented Abstraction: Object Oriented Programming Paradigm and features, Class hierarchy, Inheritance, Information hiding	Chapter 12,13,14 (T1) Chapter 7 (R1)
19-20	• Understanding of the theory and practice of Object Oriented, functional and logic programming.	Equality of Pure Lambda Terms. Substitution Revisited. Computation with Pure Lambda Terms. Programming Constructs as Lambda-Terms.The Typed Lambda Calculus. Polymorphic Types.	Chapter 14 (R1)
21-24	• The ability to write Object Oriented, functional and logic programs. • The ability to solve problems in and using Object Oriented, functional and logic programming.	Functional Programming: Introduction and basic elements of Functional Programming Functional Programming: Lists and associated Operations, Function Declaration, Higher Order Functions, Polymorphism, Data Types	Chapter 15 (T1) Chapter 8,9,10 (R1)
25-26		Logic Programming: Relations, First Order Logic, Logic Programming and Horn-Clause Programming, Unification, Deduction and Search as a strategy for deduction, Indexing, Pruning, Definite Clause Grammars.	Chapter 16(T1) Chapter 11(R1)
27-28	• Define and list the principles of concurrency. Design and develop concurrent programs. .	Concurrency and Distribution: Threads, Synchronization features (semaphores/ monitors/ locks), Shared Memory programming.	Chapter 13 (T1) Chapter 12 (R1)

Evaluation Scheme:

Component	Duration	Weightage (%)	Date & Time	Nature of Component
Programming Assignments and case study of features of a programming language	Take Home Open Book		To be announced	**20%**
Test 1	Closed Book	60min	10/9, 8.30-9.30 AM	**20%**
Test 2	Closed Book	60min	22/10, 8.30-9.30 AM	**20%**

Comprehensive	Partially Open book / Closed Book	3 hours	08/12 AN	**40%**

Chamber Consultation Hour: To be announced

Notices: All notices related to the course will be displayed on the **CSIS Notice Board**, and / or course website.

Make-up Policy:
Make ups for tests shall be granted by the I/C on prior permission and only to genuine cases with the permission of the warden concerned.
Make-up for comprehensive examination will be decided and scheduled by the Instruction Division.

INSTRUCTOR-IN-CHARGE