

Course Title	Programming Languages				
Course Code	CSE235				
Course Type	Compulsory				
Level	Bachelor (1st Cycle)				
Year / Semester	2 nd Year / 4 th Semester				
Teacher's Name	TBA				
ECTS	6	ECTS	2 Hours / 14 weeks	ECTS	1 hour / 14 weeks
Course Purpose and Objectives	To appreciate more fully the variety and diversity of high-level programming languages available, and to have specific knowledge of a few. To understand the concepts of syntax and semantics and the distinction between them. To be able to read and use formal syntactic specifications, and to have a conceptual model of the runtime behavior of programs. To be aware of the various categories of programming language facilities and of some of the possible alternative designs within these categories.				
Learning Outcomes	<p>Upon successful completion of the course, students will be able to:</p> <ul style="list-style-type: none"> • Recognize different types of programming paradigms • Apply a specific programming paradigm to solve a computational problem. • Explain different programming paradigms' implementation of data types. • Explain memory organization at program execution and compilation. • Describe the significance of implementation of various features in a computer language. • Discuss various programming language concepts: binding, scope, lifetime, parameter passing etc. 				
Prerequisites	CSE200	Co-requisites	None		
Course Content	<p>Course Contents:</p> <p>Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments.</p> <p>Evolution of the Major Programming Languages</p> <p>Describing Syntax and Semantics: Introduction, the General Problem of Describing Syntax, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs, Dynamic Semantics.</p>				

	<p>Lexical and Syntax Analysis: Lexical Analysis, The Parsing Problem, Recursive-Descent Parsing, Bottom-Up Parsing.</p> <p>Names, Bindings, and Scopes: Introduction, Names, Variables, The Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants.</p> <p>Elementary Data Types: Data objects, variables, and constants; Data types; Implementation of elementary data types; Declarations; Type checking and type conversion; Numeric data types; Enumerations; Booleans; Characters.</p> <p>Structured Data Types: Structured data objects and data types; Implementation of data structure types; Declaration and type checking for data structures; arrays; Records; Character strings; Variable-size data structures; Pointers and programmer-constructed data objects; Sets; Files and input-output.</p> <p>Expressions and Assignment Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment.</p> <p>Sub-programs and Programmer-Defined Data Types: Abstraction, encapsulation, and information hiding; Sub-programs; Type definitions; Abstract data types, Inheritance, Polymorphism.</p> <p>Sequence Control: Implicit and explicit sequence control; Sequence control within expressions; Sequence control between statements; Sub-program sequence control; Simple CALL-RETURN; Recursive Sub-programs; Coroutines; Scheduled Sub-programs;</p> <p>Data Control: Names and referencing environments; Static and dynamic scope; Block structure; Local data and local referencing environments; Shared data: Explicit common environments; Shared data: Dynamic scope; Shared data: Static scope and block structure; Shared data: Parameters and parameter transmission;</p> <p>Storage Management: Major run-time elements requiring storage; System-controlled storage management; Garbage Collection, Static storage management; Stack-based storage management; Heap storage management: Fixed-size elements; Heap storage management: Variable-size elements.</p> <p>Syntax and Translation: General syntactic criteria; Syntactic elements of a language; Stages in translation; Formal definition of syntax. BNF grammars</p> <p>Recent developments and contemporary issues pertaining to the subject-matter of the course.</p>
Teaching Methodology	Face- to- face
Bibliography	<p>Sebesta, R. W., CONCEPTS OF PROGRAMMING LANGUAGES, Addison Wesley</p> <p>Scott, M. L., PROGRAMMING LANGUAGE PRAGMATICS, Morgan Kaufmann</p> <p>Tuker & Noonan, PROGRAMMING LANGUAGES:</p>

	PRINCIPLES & PARADIGMS, Mc-Graw Hill Sethi, R., PROGRAMMING LANGUAGES: CONCEPTS & CONSTRUCTS, Addison Wesley										
Assessment	<table border="1"> <tr> <td>Mid – Term Examination</td> <td>30%</td> </tr> <tr> <td>Final Examination</td> <td>40%</td> </tr> <tr> <td>Assignments/Lab</td> <td>20%</td> </tr> <tr> <td>Class Participation and Attendance</td> <td>10%</td> </tr> <tr> <td></td> <td>100%</td> </tr> </table>	Mid – Term Examination	30%	Final Examination	40%	Assignments/Lab	20%	Class Participation and Attendance	10%		100%
Mid – Term Examination	30%										
Final Examination	40%										
Assignments/Lab	20%										
Class Participation and Attendance	10%										
	100%										
Language	English										