

ITEM	DETAILS	
1. Title of subject	PROGRAMMING LANGUAGE CONCEPTS & PARADIGMS	
2. Subject code	STC112	
3. Status of subject	Core	
4. Stage	Year 1	
5. Credit Hour	4	
6. Pre-Requisite	None	
7. Assessment	<p>40% Coursework Test 1 10% Test 2 10% Assignment 1 10% Assignment 2 10%</p> <p>60% Final Exam</p>	
8. Semester	Semester 2	
9. Objective of subject	To enable students to: <ul style="list-style-type: none"> Understand concepts and paradigms of programming languages. 	
10. Synopsis of subject	To expose the students to the various programming language concepts and paradigms. Also to introduce 2 computing languages i.e. LISP and Prolog. Background and history, semantic and syntax of language processors; Introduction to LISP and Prolog; Basic programming language concepts; Variable values and types, storage, binding, abstraction; Types system, concurrency; Introduction to programming paradigms; Imperative, concurrent, object oriented, functional, logic; Case studies on logic programming paradigm using prolog; Language selection & issues in language design.	
11. Details of subject	Contents	Hours

Week 1	<p>Background and History</p> <ul style="list-style-type: none"> • Reasons for studying concepts of programming language • Programming domains • Language evaluation criteria • Influences on language design • Language design trade off • Implementation methods • Programming environment • Evolution of major programming language <hr/> <p>Learning Outcomes: .At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand the history and evolution of programming languages and their domains. 	<p>4</p>
Week 2	<p>Describing Syntax and Semantics</p> <ul style="list-style-type: none"> • Introduction • The general problem of describing syntax • Formal method of describing syntax <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand the problems of describing syntax and solving them. 	<p>4</p>
Week 3	<p>Lexical and Syntax Analysis</p> <ul style="list-style-type: none"> • Introduction • Lexical Analysis • The parsing Problem • Recursive-Descent Parsing • Bottom-up Parsing <p>Teaching in Lab Functional Programming Paradigm</p> <ul style="list-style-type: none"> • Mathematical functions • Fundamentals of programming languages • Case study : LISP <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand recursive parsing and bottom up parsing and thereby able to differentiate them. 	<p>4</p>
Week 4	<p>Names, Binding, Type Checking and Scopes</p> <ul style="list-style-type: none"> • Introduction, names, variables • The concept of binding, type checking, strong typing • Type compatibility • Scope, scope and lifetime, referencing environment • Name constant, variable initialization <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to</p> <ul style="list-style-type: none"> • Understand the different characteristics of variables and their usage constraints in different languages. 	<p>4</p>

<p>Week 5</p>	<p>Data types</p> <ul style="list-style-type: none"> • Introduction, primitive data types, Character string types • User define ordinal type, array types, record types • Union types set type, pointer type <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand the various data types and recognise their significance and implementation. 	<p>4</p>
<p>Week 6</p>	<p>TEST 1 & TUTORIALS (CLASS & LAB)</p>	<p>4</p>
<p>Week 7</p>	<p>Expression and the Assignment Statements</p> <ul style="list-style-type: none"> • Introduction • Overloaded operators • Conversions, relational and boolean • Expressions, short circuit evaluation, Assignments Statements, Mix – mode assignments <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Construct statements and expressions using different operators and variables. 	<p>4</p>
<p>Week 8</p>	<p>Statement – Level Control Structure</p> <ul style="list-style-type: none"> • Introduction • Compound statements, Selective statements, Iterative statements, Unconditional branching, Guarded commands <p>Teaching in Lab Logic Programming Paradigm</p> <ul style="list-style-type: none"> • Introduction • A brief introduction to predicate calculus • An overview of logic programming • Case study : Prolog <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand the different types of statements and their usage. <p>Activity: Video/Peer Presentation/Debate, etc.</p>	<p>4</p>

Week 9 and 10	<p>Subprograms</p> <ul style="list-style-type: none"> • Introduction • Fundamentals of Subprograms, design issues for subprograms, local referencing environment • Parameter passing method, parameters that are subprogram names • Overloaded subprograms, Generic • Subprograms, separate and independent • Compilation, design issues for functions • Accessing non-local environment, user-defined • Overloaded operator • Coroutines <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand Subprograms and their design issues. 	<p>8</p>
Week 11 and Week 12	<p>Implementing Subprograms</p> <ul style="list-style-type: none"> □ The General Semantics of Calls and Returns □ Implementing FORTRAN 77 Subprograms □ Implementing Subprograms in ALGOL-like Languages □ Blocks □ Implementing Dynamic Scoping □ Implementing Parameters that are Subprogram Names <p>TEST 2</p> <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand subprogram implementation. <hr/> <p>Activity: Video/Peer Presentation/Debate, etc.</p>	<p>8</p>
Week 13 and Week 14	<p>Abstract Data Type</p> <ul style="list-style-type: none"> • Concept of abstraction, encapsulation, introduction to data encapsulation, design issues, language examples • Parameterized abstract data type <p>Object-Oriented Programming Paradigm</p> <ul style="list-style-type: none"> • Object-oriented programming • Inheritance, Polymorphism and Dynamic Binding • Design Issues for Object-oriented Programming • Implementation and Interface Inheritance • Single and Multiple Inheritance • Allocation and De-allocation of Objects <hr/> <p>Learning Outcomes: At the end of this lesson, students will be able to:</p> <ul style="list-style-type: none"> • Understand concepts of object orientation like data abstraction , inheritance, Polymorphism and binding. 	<p>8</p>
	<p>Total</p>	<p>56</p>

12. Text	Compulsory	<ol style="list-style-type: none">1. Sebesta, R. W. (2003). <i>Concepts of Programming Languages</i> (6th ed.). Addison-Wesley Publishing Company.2. Spivey, M. (1996). <i>Logic Programming: Essence of Prolog</i>. New Jersey: Prentice Hall.
	Reference	<ol style="list-style-type: none">1. Rawla, R. R., & Hagen, M. A. (1998). <i>Autolisp Programming Principles and Techniques</i>. US: Goodheart-Willcox Co.