Title (Units): **COMP4066 Principles of Programming Language (3,2,1)**

Course Aims: To make students understand why language features are designed the way they are. To make students capable to choose an appropriate language for a given application, to learn new languages easily, and to make clear and efficient use of any given language.

Prerequisite: COMP2045 Programming and Problem Solving AND
COMP2046 Problem Solving Using Object Oriented Approach

Course Intended Learning Outcomes (CILOs):
Upon successful completion of this course, students should be able to:

No.	Course Intended Learning Outcomes (CILOs)
	Knowledge
1	Explain the language features and paradigms of different programming languages
2	Describe syntax, semantics, names, scopes and bindings
3	Describe control flow, data types and execution of programming languages
	Professional Skill
4	Determine appropriate languages for given applications
5	Apply programming languages effectively
	Attitude
6	Engage in lifelong learning with new programming languages and paradigms

Calendar Description: This course introduces the concepts that underline most of the programming languages students are likely to encounter, and illustrates those concepts with examples from various languages. Topics include syntax and semantic analysis, bindings, type systems, programming paradigms, control abstraction and flow, and runnable program buildup.

Teaching and Learning Activities (TLAs):

CILOs	Type of TLA
1-6	Students will attend lectures to learn the principles of programming language.
4-6	Students will work on written assignments to enhance their understanding on the principles of programming language.

Assessment:

No.	Assessment Methods	Weighting	CILOs to be addressed	Description of Assessment Tasks
1	Continuous Assessment	30%	1-6	Continuous assessments are designed to measure how well students have learned the basic concepts in language features and paradigms of different programming languages.
2	Examination	70%	1-5	Final examination questions are designed to see how far students have achieved their intended learning outcomes. Questions will primarily be analysis and skills based to assess students' awareness of the features and paradigms of different programming languages.

Assessment Rubrics:

	Excellent (A)	Good (B)	Satisfactory (C)	Marginal Pass (D)	Fail (F)
language features and	Demonstrates thorough	Demonstrates sufficient	Demonstrates moderate	Demonstrates some knowledge	Demonstrates limited

paradigms of different programming languages	knowledge and understanding of language features and paradigms of different programming languages	knowledge and understanding of language features and paradigms of different programming languages	knowledge and understanding of language features and paradigms of different programming languages	and understanding of language features and paradigms of different programming languages	knowledge and understanding of language features and paradigms of different programming languages
syntax, semantics, names, scopes and bindings	Demonstrates thorough knowledge and understanding of key concepts of syntax, semantics, names, scopes and bindings	Demonstrates sufficient knowledge and understanding of key concepts of syntax, semantics, names, scopes and bindings	Demonstrates moderate knowledge and understanding of key concepts of syntax, semantics, names, scopes and bindings	Demonstrates some knowledge and understanding of key concepts of syntax, semantics, names, scopes and bindings	Demonstrates limited knowledge and understanding of key concepts of syntax, semantics, names, scopes and bindings
control flow, data types and execution of programming languages	Demonstrates thorough knowledge and understanding of key concepts of control flow, data types and execution of programming languages	Demonstrates sufficient knowledge and understanding of key concepts of control flow, data types and execution of programming languages	Demonstrates moderate knowledge and understanding of key concepts of control flow, data types and execution of programming languages	Demonstrates some knowledge and understanding of key concepts of control flow, data types and execution of programming languages	Demonstrates limited knowledge and understanding of key concepts of control flow, data types and execution of programming languages
Proficiency of using programming languages	Has a high degree of effectiveness and correctness in determining appropriate languages for given applications and applying programming languages	Has a considerable degree of effectiveness and correctness in determining appropriate languages for given applications and applying programming languages	Has a moderate degree of effectiveness and correctness in determining appropriate languages for given applications and applying programming languages	Has some degree of effectiveness and correctness in determining appropriate languages for given applications and applying programming languages	Has a limited degree of effectiveness and correctness in determining appropriate languages for given applications and applying programming languages

Course Content and CILOs Mapping:

Content		**CILO No.**
I	Introduction	1,6
II	Syntax and Semantic Analysis	2,4
III	Names, Scopes, and Bindings	2,5
IV	Control Flow	3,5
V	Data Types	3,4,5
VI	Subroutines and Control Abstraction	3,4,5
VII	Building a Runnable Program	3,4,5
VIII	Programming Paradigms	1,4,5,6

References:

- Michael L. Scott, Programming Language Pragmatics, 4th Edition, Morgan Kaufmann, 2015.
- T.W. Pratt, and M.V. Zelkowitz, Programming Languages: Design and Implementation, 4th Edition, Prentice-Hall, 2001.
- Daniel P. Friedman and Mitchell Wand, Essentials of Programming Languages, 3rd Edition, MIT Press, 2008

- Robert W. Sebesta, Concepts of Programming Language, 11th Edition, Pearson International, 2015.

Course Content:

Topic

I. Introduction
   A. Programming linguistics
   B. Historical background

II. Syntax and Semantic Analysis
   A. Specifying syntax
   B. Scanning and parsing
   C. Theoretical foundations
   D. Attribute grammars and flow
   E. Action routines
   F. Semantic functions

III. Names, Scopes, and Bindings
   A. The notion of binding time
   B. Object lifetime and storage management
   C. Scope rules
   D. Overloading, polymorphism and related concepts
   E. The binding of referencing environments

IV. Control Flow
   A. Expression evaluation
   B. Structured and unstructured flow
   C. Sequencing and selection
   D. Iteration and recursion

V. Data Types
   A. Type systems
   B. Type checking
   C. Records and variants
   D. Arrays

VI. Subroutines and Control Abstraction
   A. Calling sequences
   B. Parameter passing
   C. Generic subroutines and modules
   D. Exception handling

VII. Building a Runnable Program
   A. Back-end compiler structure
   B. Intermediate forms
   C. Code generatio

VIII. Programming Paradigms
   A. Object-oriented programming
   B. Functional and logic languages
   C. Concurrent programming