

University of Computer Studies, Yangon

B.C.Sc (Third Year)

Course code number	CS-306	Course Title	Advanced Programming Techniques
Semester hours	3 hours	No. of Credit Units	3
Prerequisite	None	Course Coordinator	Daw Soe Soe Min

Course Aims

This course will cover fundamental concepts of the majority of many programming languages: techniques for syntax and semantic analysis of programming languages and the major constructs and concepts of procedure, functional and logic languages. It aims to provide not only a unified view of many programming languages, but also the foundation which makes it easier for students to grasp/evaluate new languages and enables better programming (modeling) skills.

Learning Outcomes:

Students who have completed this course should:

- be able to distinguish between different types of programming languages such as imperative, object oriented, scripting and web languages
- be able to recognize regular expressions used in lexical analysis and BNF and parse trees used in syntax analysis.
- be able to discuss issues in imperative language design including types, binding, lifetime, storage, scope, functions, and sequencing
- be able to explain the conceptual basis of object-oriented programming languages and outline particular features of examples of them
- be able to develop small programs using various programming languages
- be able to understand the main conceptual features of modern programming languages

Course Contents:

This course provides students with a broad foundation in computer science.

- Preliminaries: Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, language design trade-offs and implementation methods.
- Evolution of the Major Programming Languages: Combining Imperative and Object-Oriented Features: C++, An Imperative-Based Object-Oriented Language: java, Scripting Languages, the Flagship .Net Language: C# and Markup/ Programming Hybrid Languages.
- Describing Syntax and Semantics: The general problem of describing syntax, formal methods of describing syntax, attribute grammars, describing the meanings of programs: Dynamic semantics
- Lexical and Syntax Analysis: The Parsing problem about top-down parsers and bottom-up parsers and describe about the complexity of parsing, and recursive-descent parsing, bottom-up parsing

- Names, Bindings and Scopes: Names, Variables, the concept of binding, scope, lifetime and referencing environments
- Data Types: Primitive Data Types, Character string type, user-defined ordinal types, and Array type describes with design issues, arrays and indices, subscript binding and array categories, array initialization, array operations, rectangular and jagged arrays and implementation of array types, Association arrays, record types and pointer and reference types, type checking, string typing and type equivalence.

Reference Book:

- (1) “The Essence of Compilers”, written by Robin Hunter
- (2) “Concepts of Programming Languages”, 10th edition by Robert W. Sebesta

Course Organization

Student participation in the course will involve the following activities:

1. Attending the lectures
2. Preparing for and participating in the recitations
3. Assignments
4. Reading Assignment
5. Quiz
6. Exam

Assessment plan for the course

Exam	60 %
Test /Assignment	10 %
Tutorial/ Practical	10 %
Moodle Test / Quiz	10 %
Class participation	10 %

Period : 45 Period of 15 weeks (50min for 1 period)

No.	Chapter	Page	Period	Detailed Lecture
	Chapter1 Preliminaries	2-30	4	
1.	1.1 Reasons for Studying Concepts of Programming Languages 1.2 Programming Domains 1.3 Language Evaluation Criteria		1	Overview Overview Explain in details
2.	1.4 Influences on Language Design 1.5 Language Categories 1.6 Language Design Trade-Offs		1	Overview Overview Explain in details
3.	1.7 Implementation Methods		1	Explain in details
4.	Review Questions		1	3-10,13-18,22-28,30
5.	Practical Exercise			
	Chapter 2 Evolution of the Major Programming Languages	88-106	3	
6.	2.16 Combining Imperative and Object-Oriented Features: C++ 2.17 An Imperative-Based Object-Oriented Language: Java 2.18 Scripting Languages 2.19 The Flagship .NET Language: C# 2.20 Markup/Programming Hybrid Languages		3	Reading assignment and Discuss language features with students
	Chapter 3 Describing Syntax and Semantics	114-149	8	

7.	3.1 Introduction 3.2 The General Problem of Describing Syntax 3.3 Formal Methods of Describing Syntax		3	Overview Overview Explain in details
8.	3.4 Attribute Grammars 3.5 Describing the Meanings of Programs: Dynamic Semantics		2	Overview Overview
9.	Review Questions		1	1-10,12,15-16,28
10.	Problem Sets		1	3-18
11.	Practical Exercises using Tool (JFLAP Tool)		1	
	Chapter 4 Lexical and Syntax Analysis	168-197	14	
12.	4.1 Introduction 4.2 Lexical Analysis		2	Overview Explain in details
13.	4.3 The Parsing Problem 4.3.1 Introduction to Parsing 4.3.2 Top-Down Parsers 4.3.3 Bottom-Up Parsers 4.3.4 The Complexity of Parsing		4	Explain in details Explain in details Explain in details Explain in details Overview
14.	4.4 Recursive-Descent Parsing 4.5 Bottom-Up Parsing		5	Explain in details Explain in details
15.	Review Questions		1	1-6,8-11,18,23
16.	Problem Sets		1	1-10
17.	Practical Exercises using Tool (JFLAP Tool)		1	Programming Exercise 1-4
	Chapter 5 Names, Bindings, and Scopes	204-232	7	
18.	5.1 Introduction 5.2 Names		1	Overview Explain in details
19.	5.3 Variables 5.4 The Concept of Binding 5.5 Scope 5.6 Scope and Lifetime 5.7 Referencing Environments		4	Explain in details Explain in details Overview Overview Explain in details
20.	Review Questions		1	1-2,4,6-13,15-17,22
21	Problem Sets Programming Exercise		1	1,4-12 5
	Chapter 6 Data Types	244-308	9	
22.	6.1 Introduction 6.2 Primitive Data Types 6.3 Character String Types 6.4 User-Defined Ordinal Types		2	Overview Explain in details Explain in details Overview

23.	6.5 Array Types 6.5.1 Design Issues 6.5.2 Arrays and Indices 6.5.3 Subscript Bindings and Array Categories 6.5.4 Array Initialization 6.5.5 Array Operations 6.5.6 Rectangular and Jagged Arrays 6.5.9 Implementation of Array Types 6.6 Associative Arrays 6.7 Record Types		2	Explain in details Overview Explain in details Overview Overview Explain in details Explain in details Overview Overview Overview
24.	6.11 Pointer and Reference Types 6.11.1 Design Issues 6.11.2 Pointer Operations 6.11.3 Pointer Problems 6.11.4 Pointers in Ada 6.11.5 Pointers in C and C++ 6.11.6 Reference Types 6.11.7 Evaluation 6.11.8 Implementation of Pointer and Reference Types		2	Explain in details Overview Overview
25.	6.12 Type Checking 6.13 Strong Typing 6.14 Type Equivalence		1	Explain in details Overview Overview
26.	Review Questions		1	1-9,12-13,15,17-18,20,22,23,35-41,43-46,48-53
27.	Revision		1	