Course Specifications

Course Title:	Concepts of Programming Language
Course Code:	**IT326**
Program:	**Computer Science and Information Technologies**
Department:	**Computer Science and Information**
College:	**College of Science at Az Zulfi**
Institution:	**Majmaah University**

Table of Contents

A. Course Identification

1. Credit hours: 3
2. Course type **a.** University ☐ College ☐ Department ☐ Others ☐ **b.** Required ☐ Elective ☐
3. Level/year at which this course is offered: elective
4. Pre-requisites for this course (if any)**:**
5. Co-requisites for this course (if any)**:**

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	**Traditional classroom**		
2	**Blended**		
3	**E-learning**		
4	**Correspondence**		
5	**Other**		

7. Actual Learning Hours (based on academic semester)

No	Activity	Learning Hours
Contact Hours		
1	**Lecture**	
2	**Laboratory/Studio**	
3	**Tutorial**	
4	**Others** (specify)	
	Total	
Other Learning Hours*		
1	**Study**	
2	**Assignments**	
3	**Library**	
4	**Projects/Research Essays/Theses**	
5	**Others** (specify)	
	Total	

* The length of time that a learner takes to complete learning activities that lead to achievement of course learning outcomes, such as study time, homework assignments, projects, preparing presentations, library times

B. Course Objectives and Learning Outcomes

1. Course Description

This course introduces students to the main constructs of contemporary programming languages and provides the tools needed to evaluate the existing and future programming languages. Topics include: Reasons for Studying Concepts of Programming Languages, Evolution of the Major Programming Languages, Describing Syntax and Semantics, Names, Bindings, and Scopes Data Types Expressions and Assignment Statements, Statement-Level Control Structures, Subprograms

2. Course Main Objective

The purpose of this course is to :
1. Provide students with better understanding of significance of implementation.
2. Acquaint students with the ability to learn new languages.
3. Introduce students to various programming language concepts including binding, scope, lifetime, parameter passing etc.
4. Enable students to be efficient in their work.

3. Course Learning Outcomes

CLOs		Aligned PLOs
1	**Knowledge:**	
1.1	Understand the fundamental programming constructs including Names, Bindings, Scopes, Data Types, Expressions and Assignment Statements, Statement-Level Control Structures, subprograms.	
1.2	Name the key programming language concepts	
2	**Skills :**	
2.1	Improved background for choosing appropriate languages	
2.2	Increased ability to learn new languages	
3	**Competence:**	
3.1	Better understanding of significance of implementation	
3.2	Better use of programming languages that are already known	

C. Course Content

No	List of Topics	Contact Hours
1	Concepts of Programming Languages, Programming Domains ains, Language Evaluation Criteria, Influences on Language Design sign, Language Categories, Language Design Trade-Offs, Implementation .Methods, Programming Environments	6
2	Evolution of the Major Programming Languages : Zuse's Plankalkül, Pseudocodes, Fortran, LISP, ALGOL 60, COBOL, BASIC, Object-Oriented Programming, C++, Java, Scripting Languages, .NET Language: C#, Markup / Programming Hybrid Languages .	6
3	(Operational Semantics) 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.	6

4	Names, Bindings, and Scopes: Introduction, Names, Variables, The Concept of Binding, Scope, Named Constants.	6
5	Data Types: Introduction, Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Associative Arrays, Record Types.	6
6	Expressions and Assignment Statements: Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements,	6
7	Statement-Level Control Structures: Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands, Conclusions.	3
8	Subprograms: Introduction, Fundamentals of Subprograms, Design Issues for Subprograms, Local Referencing, Environments, Parameter-Passing Methods.	3
9	Course conclusion	3
Total		

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge**		
1.1	Understand the fundamental programming constructs including Names, Bindings, Scopes, Data Types, Expressions and Assignment Statements, Statement-Level Control Structures, subprograms.	Lectures Lab demonstrations Case studies Individual presentations	Written Exam Homework assignments Lab assignments Class Activities Quizzes
1.2	Students will be able to learn new programming paradigms and languages on their own.		
2.0	**Skills**		
2.1	Improved background for choosing appropriate languages	Lectures Lab demonstrations Case studies Individual presentations Brainstorming	Written Exam Homework Assignments Lab assignments Class Activities Quizzes
2.2	Students will be able to describe programming language syntax formally and semantics informally.		
3.0	**Competence**		
3.1	Better understanding of significance of implementation	Small group discussion Whole group discussion Brainstorming Presentation	Written Exam Homework assignments Lab assignments Class Activities Quizzes
3.2	Better use of programming languages that are already known		

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	First written mid-term exam	6	20%
2	Second written mid-term exam	12	20%

#	Assessment task*	Week Due	Percentage of Total Assessment Score
3	Presentation, class activities, lab activity, and group discussion	Every week	10%
4	Homework assignments	After every chapter	10%
5	Final written exam	15	40%
	TOTAL		100%

***Assessment task** (i.e., written test, oral test, oral presentation, group project, essay, etc.)

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice :

1. A total of 6 office hours per week in the lecturer schedule in order to facilitate the student.
2. Contacting students using e-mail , mobile, office telephone and website.

F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	• RAMSEY, Norman. Programming Languages: Build, Prove, and Compare. 2016.
Essential References Materials	• David A. Watt, Programming Language Design Concepts, Wiley (May 31, 2004). • Sara Baase and Allen Van Gelder, Computer Algorithms: Introduction to Design & Analysis, Third Edition. by, 2000.
Electronic Materials	
Other Learning Materials	Video and presentation are available with me

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Classroom and Lab, as those that are available at college of science at AzZulfi
Technology Resources (AV, data show, Smart Board, software, etc.)	Smart Board
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	None

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of Teaching	Students	• Analysis of students' results. Observation during class work. • Students' evaluations. • Colleagues' evaluations. • Evaluation questionnaire filled by the students. Interview a sample of students enrolled in the course to take their opinions
Evaluation of Teaching	Program leaders	• Self-assessment. • External evaluation. Periodic review of course (the Commission of study plans)

Evaluation areas (e.g., Effectiveness of teaching and assessment, Extent of achievement of course learning outcomes, Quality of learning resources, etc.)
Evaluators (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)
Assessment Methods (Direct, Indirect)

H. Specification Approval Data

Council / Committee	
Reference No.	
Date	