Course Specifications

Institution:	**College of Science at Az Zulfi**
Academic Department :	**Computer Science and Information**
Programme :	**Computer Science and Information**
Course :	**Concepts of Programming Languages**
Course Coordinator :	**Noureldin Laban**
Programme Coordinator :	**Ass. Prof. Dr. Yosry Azzam**
Course Specification Approved Date :	22/ 12 / 1435 H

A. Course Identification and General Information

1 - Course title : **Concepts of Programming Languages**	Course Code: **CSI 513**

2. Credit hours : **(3) (2 Lec. + 2 Lab)**

3 - Program(s) in which the course is offered: **Computer science and information**

4 – Course Language : **English**

5 - Name of faculty member responsible for the course: **Noureldin Laban**

6 - Level/year at which this course is offered : : **9th level – 1435/1436**

7 - Pre-requisites for this course (if any) :
- **Discrete Mathematics for Computer Science 2 (CSI 222)**

8 - Co-requisites for this course (if any) :
-

9 - Location if not on main campus **:**
(....)

10 - Mode of Instruction (mark all that apply)

A - Traditional classroom	√	What percentage?	**80 %**
B - Blended (traditional and online)	√	What percentage?	**10 %**
D - e-learning	√	What percentage?	**10 %**
E - Correspondence		What percentage?	……. %
F - Other		What percentage?	……. %

Comments :
....

B Objectives

What is the main purpose for this course?

Concepts of Programming Languages introduces students to the main constructs of contemporary programming languages and provides the tools needed to critically evaluate the existing and future programming languages. Students gain a solid foundation for understanding the fundamental concepts of programming languages through the course presentation of design issues for various language constructs, the examination of the design choices for these constructs in some of the most common languages, and critical comparison of the design alternatives. In addition, the course provides an in-depth discussion of programming language structures, presents a formal method of describing syntax, and introduces approaches to lexical and syntactic analysis.
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.

Briefly describe any plans for developing and improving the course that are being implemented :

1. Using group discussions through the internet with course attending students.
2. Updating the materials of the course to cover the new topics of the field.
3. Increasing the ability of the students to implement the concepts that are presented in the course.

C. Course Description

1. Topics to be Covered

List of Topics	No. of Weeks	Contact Hours
1. Preliminaries: 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.	2	8
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 .	1	4
3. 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.	3	12
4. Names, Bindings, and Scopes: Introduction, Names, Variables, The Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants.	2	8
5. Data Types: Introduction, Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Associative Arrays, Record Types.	2	8
6. Expressions and Assignment Statements: Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment.	2	8
7. Statement-Level Control Structures: Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands, Conclusions.	2	8
8. Subprograms: Introduction, Fundamentals of Subprograms, Design Issues for Subprograms, Local Referencing, Environments, Parameter-Passing Methods.	1	4
9. Expressions and Assignment Statements: Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment.	1	4

2. Course components (total contact hours and credits per semester):

	Lecture	Tutorial	Laboratory	Practical	Other:	Total
Contact Hours	**30**	....	....	**30**	....	60
Credit	**30**	....	....	**15**	....	45

3. Additional private study/learning hours expected for students per week.

5

Total Hours =   60+ 75 = 135 hour s

4. Course Learning Outcomes in NQF Domains of Learning and Alignment with Assessment Methods and Teaching Strategy

	NQF Learning Domains And Course Learning Outcomes	Course Teaching Strategies	Course 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	**Name the key programming language concepts**		
2.0	**Cognitive Skills**		
2.1	**Improved background for choosing appropriate languages**	Lectures Lab demonstrations	Written Exam Homework assignments
2.2	**Increased ability to learn new languages**		

	NQF Learning Domains And Course Learning Outcomes	Course Teaching Strategies	Course Assessment Methods
		Case studies Individual presentations Brainstorming	Lab assignments Class Activities Quizzes
3.0	**Interpersonal Skills & Responsibility**		
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**		
4.0	**Communication, Information Technology, Numerical**		
4.1	**Work cooperatively in a small group.**	Small group discussion Whole group discussion Brainstorming Presentation	Written Exam Homework assignments Lab assignments Class Activities Quizzes
4.2	**Save time and space in each task.**		
5.0	**Psychomotor**		

5. Schedule of Assessment Tasks for Students During the Semester:

	Assessment task	Week Due	Proportion of Total Assessment
1	First written mid-term exam	6	15%
2	Second written mid-term exam	12	15%
3	Presentation, class activities, and group discussion	Every week	10%
4	Homework assignments	After Every chapter	10%
5	Implementation of presented concepts	Every two weeks	10%
6	Final written exam	16	40%

D. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice. (include amount of time teaching staff are expected to be available each week)

Office hours: Sun: 8-12, Mon. 10-12, Wed. 8-10
Office call: Sun. 12-1 and Wed 12-1

Email: n.laban@mu.edu.sa

E. Learning Resources

1. List Required Textbooks :
- Robert W. Sebesta, Concepts of Programming languages, Addison-Wesley edition 10, 2013.

2. List 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.

3. List Recommended Textbooks and Reference Material :
- **Journal of Computer Languages.**

4. List Electronic Materials :
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-821-programming-languages-fall-2002/index.htm
- http://nptel.ac.in/courses/106102067/

5. Other learning material :
- **Video and presentation are available with me**

F. Facilities Required

1. Accommodation
Classroom and Lab, as those that are available at college of science at AzZulfi.

2. Computing resources
- **Smart Board**

3. Other resources
- **N/A**

G   Course Evaluation and Improvement Processes

1 Strategies for Obtaining Student Feedback on Effectiveness of Teaching:
- Questionnaires (course evaluation) filled by the students and electronically organized by the university.
- Student-faculty management meetings.

2 Other Strategies for Evaluation of Teaching by the Program/Department Instructor :
- Discussion within the staff members teaching the course
- Departmental internal review of the course.

3 Processes for Improvement of Teaching :
- Periodical departmental revision of methods of teaching.
- Monitoring of teaching activities by senior faculty members.
- Training courses.

4. Processes for Verifying Standards of Student Achievement
- Evaluation matrix

5 Describe the planning arrangements for periodically reviewing course effectiveness and planning for improvement :
- Course evaluation
- Exam evaluation
- Improvement plan

Course Specification Approved
Department Official Meeting No (6) Date 22 / 12 / 1435 H

Course's Coordinator

Name : Noureldin Laban

Signature : ....

Date : 17/ 12 / 1435 H

Department Head

Name : Dr. Yosry Azzam

Signature : ....

Date : 22/ 12 / 1435 H