University/Academy: Arab Academy for Science and Technology & Maritime Transport
Faculty/Institute: College of Computing and Information Technology
Program: Computer Science

Form No. (12)
Course Specification

1- Course Data

Course Code:	Course Title:	Academic Year/Level:
CS345	Structure of programming languages	Year 2 / Semester 4
Specialization: Computer Science	**No. of Instructional Units:** 2 hrs lecture 2 hrs lab	**Lecture:**

2- Course Aim	A concise introduction to the essentials of different programming languages (Imparative, OOP, functional, and Logic programming), focusing on principles rather than specifics. Fundamental issues in language design. Overview of programming paradigms. Type systems: Data types; type constructors, type compatibility, type conversions. Models of execution control: Order of evaluation of subexpressions; conditional execution; iteration; subprogram unit, passing parameters and Implementation of subprogram units..
3- Intended Learning Outcome:	
a- Knowledge and Understanding	**Students will be able to demonstrate knowledge of:** K10. Current developments in computing and information research. K13. Use high-level programming languages. K18. Understand the fundamental topics in Computer Science, including hardware and software architectures, software engineering principles and methodologies, operating systems, compilers, parallel and distributed computing, systems and software tools. 1. Know the aims of the course 2. Know the reasons for studying concepts of programming languages 3. Know the programming environments 4. Understand the general problem of describing syntax 5. Describe the formal methods of describing syntax 6. Understand the attribute grammars 7. Understand the lexical analysis 8. Describe the parsing problem 9. Understand the recursive descent parsing 10. Understand the bottom up parsing 11. Understand various variable attributes and the concepts of binding 12. Differentiate static and dynamic scoping 13. Describe referencing environments 14. Understand the primitive data types, string types, user-defined ordinal types, array types, record and union types, pointers and reference types

	15. Understand the type checking, strong typing, and type compatibility 16. Write arithmetic and Boolean expressions 17. Understand precedence and associativity 18. Know the type conversions and mixed-mode assignments 19. Describe Selection Statements 20. Describe Iterative Statements 21. Learn Fundamentals of Subprograms 22. Understand how Subprograms could be passed as parameters 23. Define User-Defined Overloaded Operators 24. Learn a new programming language
b- Intellectual Skills	**By the end of the course, the student acquires high skills and an ability to understand:** I10. Define traditional and nontraditional problems, set goals towards solving them, and. observe results. I11. Perform comparisons between (algorithms, methods, techniques...etc). I17. Identify a range of solutions and critically evaluate and justify proposed design solutions. 1. Develop a finite automata for a subset of a language grammar 2. Analyze variable scopes 3. Use different data types in problem abstraction 4. Evaluate short-circuit expressions 5. Analyze operator overloading 6. Evaluate Unconditional Branching 7. Analyze the design issues of subprograms 8. Analyze Design Issues for Functions 9. Analyze concepts of a new programming language

c- Professional Skills	**By the end of the course the student will have the ability to:** P15. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem. P19. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. 1. Write a lexical analyzer 2. Write recursive descent parsing routines 3. Check variable scopes and referencing environments 4. Write various types of expressions in different programming languages 5. Write structured programs 6. Develop a complete application in a new programming language
d- General Skills	**Students will be able to:** G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G2. Demonstrate skills in group working, team management, time management and organizational skills. G7. Show the use of general computing facilities. 1. Be able to think critically 2. Learn about problem abstraction 3. Appreciate structured programming

4- Course Content	

1	Be familiar with several language paradigms and how they relate to different application domains.
2	Understand the design space of programming languages, including concepts and constructs from past languages as well as those that may be used in the future.
3	Understanding of the programming language we use by being able to identify and compare the same concept as it appears in different languages.
4	Evaluating of programming models to provide a range of possible solutions and the ability to select the most optimized and relevant to the problem in hand.
5	Understand the concepts and theory behind the implementation of high level programming languages
6	Know significant details about a number of important techniques commonly used in compilers construction.

5- Teaching and Learning Methods	Lectures, Labs, Projects, Individual study & self-learning.
6- Teaching and Learning Methods for Students with Special Needs	• Students with special needs are requested to contact the college representative for special needs (currently Dr Hoda Mamdouh in room C504) • Consulting with lecturer during office hours. • Consulting with teaching assistant during office hours. • Private Sessions for redelivering the lecture contents. • For handicapped accessibility, please refer to program specification.
7- Student Assessment:	
a- Procedures used:	Exams and Individual Projects
b- Schedule:	7th week exam 30% 12th week exam 20% Lab 10% Final exam 40%
c- Weighing of Assessment:	**Week 7 Grades – 30%** **Week 12 -Grades – 20%** Lab 10% **Week 16 - Final Exam – 40%**

8- List of References:

a- Course Notes	From the Moodle on www.aast.edu
b- Required Books (Textbooks)	R. Sebesta, *Concepts of Programming Language*, 7thEd., Pearson, 2005
c- Recommended Books	Pratt &Zelkowitz ,*Programming Languages: Design and Implementation*, 4/e, 2001
d- Periodicals, Web Sites, …, etc.	

Course Instructor: **Head of Department:**

Sign **Sign**