**Course Specification**

**(CS 317 : Concepts of Programming )**

|  |  |
| --- | --- |
| *University:* | Helwan University |
| *Faculty:* | Faculty of Computers & Information |
| *Department:* | ***Computer science*** |

**1. Course Data**

|  |  |
| --- | --- |
| **Code:** | **CS 317** |
| **Course title:** | **Concepts of Programming** |
| **Level:** | 1 |
| **Specialization:** | Computer Science |
| **Credit hours:** | 3 hours |
| **Number of learning units (hours):** | (3) theoretical |

**2. Course Objective**

Acquainting the students with principles and concepts of different programming languages paradigms including design and implementation issues of different models of programming including structured programming , object-oriented programming, functional and logic programming.

**3. Intended Learning Outcomes:**

* **Knowledge and Understanding:**

A7. Define the basics of Computer Systems.

A8. Apply Programming to solve Problems.

A17. Represent essential knowledge of Translators Design.

* **Intellectual Skills**

B4. Formulate and test Concepts and Hypothesis.

B8. Gather and assess relevant information, using abstract ideas to interpret it effectively.

B10. Distinguish Diagnosis Techniques.

B18. Classify different problems.

* **Professional and Practical Skills**

C1. Choose the appropriate Programming Language.

* **General and Transferable Skills**

D4. Follow Analytical Thinking.

D14. Practice Engineering skills for software development

**4. Course contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **No. of hours** | **Lecture** | **Tutorial/ Practical** |
| Introduction  Evolution of Major Programming Languages | 3 | 1 | 1 |
| Lexical and Syntax Analysis   * Describing Syntax, formal definition of languages, BNF and context-free grammars * Parsing | 6 | 2 | 1 |
| Names, Bindings, and Scopes   * Names * Variables * The Concept of Binding * Scope * Scope and Lifetime * Referencing Environments * Named Constants | 3 | 1 | 1 |
| Data Types   * Primitive Data Types * Character String Types * User-Defined Ordinal Types * Array Types * Associative Arrays * Record Types * Union Types * Pointer and Reference Types | 3 | 1 | 1 |
| Expressions and Assignment Statements   * Arithmetic Expressions * Overloaded Operators * Type Conversions * Relational and Boolean Expressions * Short-Circuit Evaluation * Assignment Statements | 3 | 1 | 1 |
| Statement-Level Control Structures   * Selection Statements * Iterative Statements * Unconditional Branching * Guarded Commands | 6 | 2 | 2 |
| Subprograms   * Design Issues for Subprograms * Local Referencing Environments * Parameter-Passing Methods * Parameters That Are Subprograms * Overloaded Subprograms * Generic Subprograms * Stack-Dynamic Local Variables * Nested Subprograms * Dynamic Scoping | 6 | 2 | 1 |
| Abstract Data Types and Encapsulation Concepts   * Data Abstraction * Design Issues for Abstract Data Types * Language Examples * Parameterized Abstract Data Types * Encapsulation Constructs * Naming Encapsulations | 6 | 2ss | 1 |
| * Object-Oriented Programming * Examples from different OO languages such as C++, Java, C#,… * Implementation of Object-Oriented Constructs | 3 | 1 | 1 |
| Overview of other types of programming   * Functional programming * Logic programming * Concurrent programming | 6 | 2 | 2 |

**Mapping contents to ILOs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topic | Intended Learning Outcomes (ILOs) | | | |
| Knowledge and understanding | Intellectual Skills | Professional and practical skills | General and Transferable skills |
| Introduction  Evolution of Major Programming Languages | A7,A8,A17 | B4 | C1 | D4 |
| Lexical and Syntax Analysis   * Describing Syntax, formal definition of languages, BNF and context-free grammars * Parsing | A7 | B8 | C1 | D14 |
| Names, Bindings, and Scopes   * Names * Variables * The Concept of Binding * Scope * Scope and Lifetime * Referencing Environments * Named Constants | A7 | B8 | C1 | D4 |
| Data Types   * Primitive Data Types * Character String Types * User-Defined Ordinal Types * Array Types * Associative Arrays * Record Types * Union Types * Pointer and Reference Types | A8 | B10 | C1 | D14 |
| Expressions and Assignment Statements   * Arithmetic Expressions * Overloaded Operators * Type Conversions * Relational and Boolean Expressions * Short-Circuit Evaluation * Assignment Statements | A17, A7 | B18 | C1 | D4 |
| Statement-Level Control Structures   * Selection Statements * Iterative Statements * Unconditional Branching * Guarded Commands | A8, A7 | B4 | C1 | D14 |
| Subprograms   * Design Issues for Subprograms * Local Referencing Environments * Parameter-Passing Methods * Parameters That Are Subprograms * Overloaded Subprograms * Generic Subprograms * Stack-Dynamic Local Variables * Nested Subprograms * Dynamic Scoping | A17,A7 | B8 | C1 | D4 |
| Abstract Data Types and Encapsulation Concepts   * Data Abstraction * Design Issues for Abstract Data Types * Language Examples * Parameterized Abstract Data Types * Encapsulation Constructs * Naming Encapsulations | A7, A7 | B10 | C1 | D14 |
| * Object-Oriented Programming * Examples from different OO languages such as C++, Java, C#,… * Implementation of Object-Oriented Constructs | A17, A8 | B18 | C1 | D14 |
| Overview of other types of programming   * Functional programming * Logic programming * Concurrent programming | A17,A7,A8 | B4, B8 | C1 | D4, D14 |

**5. Teaching and Learning Methods**

Class Lectures

Highly lab-based courses

Tutorials

**6. Teaching and Learning Methods for students with limited capability**

Using data show

e-learning management tools

**7. Students Evaluation**

**a) Used Methods**

Lab exam

Assignments

Lab work

Programming projects

**b) Time**

Assessment 1: Test 1 Week 4

Assessment 2: Test 2 Week 7

Assessment 3: Midterm Exam Week 10

Assessment 4: Practical Exam Week 14

Assessment 5: final written exam Week 16

**c) Grades Distribution**

Mid-term Examination 20 %

Final-Year Examination 50 %

Semester Work 20 %

Practical Exam 10%

Total 100%

Any formative only assessments

**List of Books and References**

**a) Notes**

Course Notes

- Handouts

**b) Mandatory Books**

Robert Sebesta, Concepts of Programming Languages, 7th Edition, Addison Wesley 2005.

**c) Suggested Books**

- [Franklyn A. HYPERLINK "http://www.amazon.com/Franklyn-A.-Turbak/e/B001J3NUBS/ref=ntt\_athr\_dp\_pel\_1"Turbak](http://www.amazon.com/Franklyn-A.-Turbak/e/B001J3NUBS/ref=ntt_athr_dp_pel_1) , [David K. Gifford](http://www.amazon.com/s/ref=ntt_athr_dp_sr_2?_encoding=UTF8&sort=relevancerank&search-alias=books&field-author=David%20K.%20Gifford), Design Concepts in Programming Languages , MIT Press , 2008

- [Peter Van Roy](http://www.info.ucl.ac.be/~pvr) and [SeifHYPERLINK "http://www.sics.se/~seif" HYPERLINK "http://www.sics.se/~seif"Haridi](http://www.sics.se/~seif), Concepts , Techniques and models of Computer Programming, MIT Press, 2004.

Periodicals, Web Sites, … etc

- <http://www.aw-bc.com/sebesta/>

**d) Other publications**

**Course Coordinator:**  Dr. Amal Aboutabl

**Chairman of the Department:** Prof. dr. Iraqy Khalifa