



---

## Assembly Language Course Specifications

**Faculty:** Computer and Informatics

**Department:** Information System

**Program(s) on which the course is given:** Bachelor in Computer and Information Sciences

**Major or Minor element of programs :** All majors

**Department offering the program :** Information System

**Department offering the course :** Computer Science

**Academic year / Level :** 3rd year / B.Sc.

**Date of specification approval :** 30/10/2009

### A. Basic Information

**Title:** Assembly Language      **Code:** CSW 353

**Lectures:** 3 hrs/week      **Practical:** 3 hrs/week      **Tutorial:** ---

**Credit Hours:** ---      **Total:** 6 hrs/week



---

## B. Professional Information

### 1. Overall Aims of Course:

To introduce the students to the architecture of a computer, and to low-level programming in assembly language for the given computer architecture.

### 2. Intended Learning Outcomes of Course (ILOs):

#### a. Knowledge and Understanding:

- a1- Describe a computer system in terms of its main components and their functionality, using real computer systems.
- a2- Understand the register level machine and basic assembly programming
- a3- Describe techniques for improving the performance and reliability of processors, memory, I/O devices and other components of a computer system.
- a4- Explain the use of parallelism, caching, error-detection and correction.
- a5- Understanding of the aspects of computer arithmetic relating to the occurrence of rounding and truncation errors and the calculation of absolute errors.
- a6- Illustrate the necessity of tradeoffs in the design of a computer system.
- a7- Be aware of the processes that drive the evolution of the modern computer system.
- a8- Explain the need for resource management in a computer system.



---

a9- Illustrate the problems caused by multiprogramming, and identify the need for process synchronization.

**b. Intellectual Skills:**

b1- Interpret binary, octal and hexadecimal numbers, and explain how such numbers are represented in a computer.

b2- Formulate, test and debug simple assembly-language programs involving the use of arithmetic expressions, loops, conditional branches and recursion

**c. Professional and Practical Skills:**

c1- Design and construct a computer-based system using simple assembly programs to solve complex real-world tasks. From observation of the behavior of these programs in the real world, evaluate and improve the original solutions.

**d. Transferable and Key Skills:**

d1- Able to solve problems effectively.

d2- Work coherently with others in a small group to solve a common problem.

**e. Attitude:**

e1- Relationship Emphasis a successful with other students.

e2- Learn how to make relation with other, and the limit of this relation.

**a. Contents:**

<b>Topic</b>	<b>No. of hours</b>	<b>Lecture</b>	<b>Practical</b>
Microprocessors: Architecture, ALU, registers, accumulators, memory addressing, control logic, addressing modes – I	5	3	2
Microprocessors: Architecture, ALU, registers, accumulators, memory addressing, control logic, addressing modes – II	5	3	2
Assembler instructions, pseudo-operations and operators, data definitions: byte, word, immediate operands, difference between EXE and COM program files – I	5	3	2
Assembler instructions, pseudo-operations and operators, data definitions: byte, word, immediate operands, difference between EXE and COM program files – II	5	3	2
Assembler instructions, pseudo-operations and operators, data definitions: byte, word, immediate operands, difference between EXE and COM program files – III	5	3	2
Assembler instructions, pseudo-operations and operators, data definitions: byte, word, immediate operands, difference between EXE and COM program	5	3	2



كلية الحاسبات و المعلومات



files – IV			
Program logic and organization: JMP, LOOP, CALL, stack segment, Boolean – I	5	3	2
Program logic and organization: JMP, LOOP, CALL, stack segment, Boolean – II	5	3	2
operation: AND, OR, XOR, TEST, NOT, shifting and rotation, string instructions – I	5	3	2
operation: AND, OR, XOR, TEST, NOT, shifting and rotation, string instructions – II	5	3	2
arithmetic: processing binary data, processing ASCII and BCD data, table processing, procedures, interrupts - I	5	3	2
arithmetic: processing binary data, processing ASCII and BCD data, table processing, procedures, interrupts - II	5	3	2
programming techniques, COM programming, numeric calculations, text management, screen processing, I/O libraries, linking with high level languages - I	5	3	2
programming techniques, COM programming, numeric calculations, text management, screen processing, I/O libraries, linking with high level languages - II	5	3	2