**University/Academy:** Arab Academy for Science, Technology & Maritime Transport
**Faculty/Institute:** College of Engineering & Technology
**Program:** B.Sc Computer Engineering

# Form no. (12): **Course Specification**

## 1- Course Data

| Course Code: | Course Title: | | Academic Year/Level: |
|---|---|---|---|
| **CC540** | **Computer Systems Engineering** | | **4th year / 7th semester** |
| Specialization: | No. of Instructional Units | Lecture | Practical |
| **Computer Engineering** | **3** | **2** | **2** |

## 2- Course Aim

- The course educates students in the integration of key topics from algorithms, computer architecture, operating systems, compilers, and software engineering, in one unified framework. Building a general-purpose computer system from the ground up. Techniques in the design of modern hardware and software systems, and major trade-offs and future trends are introduced. Throughout the course, many cross-section views of the computing field, from the bare bone details of switching circuits to the high level abstraction of object-based software design are presented.

## 3- Intended Learning Outcomes

| | |
|---|---|
| **a- Knowledge and Understanding** | **Through knowledge and understanding, students will be able to:**<br>• Define the Computer architecture.<br>• Describe the main subcategories of computer architecture.<br>• . Identify the Instruction set architecture, or ISA<br>• Describe the data paths, data processing elements, data storage elements, and describes how they should implement the ISA.<br>• Understand System Design<br>• Define operating systems (OS)<br>• Understand real-time operating system and Multi-user vs. Single-user<br>• Identify different types of OS<br>• Define general components of OS, including Kernel, Program execution, Interrupts, Modes, Memory management, and Virtual memory.<br>• Understand Software engineering/Tools/Compiler.<br>• Describe the structure of a compiler, Understand Compiled versus interpreted languages and Compiler construction.<br>• Understand Software design principles<br>• . Understand a general-purpose computer system.<br>• Understand graphics processing unit or GPU (also occasionally called visual processing unit or VPU).<br>• Understand Packet Filtering Implementations for General-Purpose Computers.<br>• Describe System Level Modeling for Hardware/Software Systems<br>• Describe System Level Language Requirements.<br>• Describe Distributed design patterns and GRASP (object-oriented design) |
| **b- Intellectual Skills** | **Through intellectual skills, students will be able to:**<br>• Classify the Computer architecture.<br>• Calculate the System Design process.<br>• Classify types of operating systems<br>• Identify a general-purpose computer system.<br>• Identify the advances in embedded systems, mobile phones, and game consoles.<br>• Classify Packet Filtering Implementations for General-Purpose Computers. |

| | |
|---|---|
| | • Classify System Level Modeling for Hardware/Software Systems and System Level Language Requirements |
| **c- Professional Skills** | **Through professional and practical skills, students will be able to:**<br>• Analyze the role of computer architecture<br>• Analyze the steps of Implementation of micro-architecture, Logic Implementation, Circuit Implementation, Physical Implementation, and Design Validation.<br>• Distinguish Microarchitecture principles.<br>• Explain the systems engineering process.<br>• Select a design pattern, a general reusable solution to a commonly occurring problem within a given context in software design.<br>• Infer Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved<br>• Analyze design patterns reside in the domain of Abstraction principle, Algorithmic skeleton, Anti-pattern, Architectural pattern, Distributed design patterns, Enterprise Architecture framework, GRASP (object-oriented design),<br>• Prepare interaction design pattern.<br>• Differentiate software development philosophies. |
| **d- General Skills** | **Through general and transferable skills, students will be able to:**<br>• Verify theory with practice<br>• Verify with practice Exercise on different models. |

## 4- Course Content

| | |
|---|---|
| **Week No.**1 | Introduction to Computer architecture, architecture components and categories of different advanced architectures. |
| **Week No.**2 | System design process & categories. |
| **Week No.**3 | Microarchitecture principles. |
| **Week No.**4 | Operating systems, components, types, design principles |
| **Week No.**5 | Compilers, and software engineering: design principles and methodology. |
| **Week No.**6 | Building general/ and special purpose computer systems. |
| **Week No.**7 | 7th Week Exam |
| **Week No.**8 | Techniques in the design of modern hardware and software systems, and major trade-offs and future trends are introduced. |
| **Week No.**9 | Cross-section views of the computing field, from the bare bone details to High level abstraction of object-based software design are presented. |
| **Week No.**10 | Building a general and special -purpose computer systems. |
| **Week No.**11 | Graphics processing units or GPUs (also occasionally called visual processing unit or VPU), a specialized circuit design to rapidly manipulate and alter memory in such a way so as to accelerate the building of images in a frame buffer intended for output to a display. |
| **Week No.**12 | 12th Week Exam. |
| **Week No.**13 | Advances in embedded systems, mobile phones, and game consoles. |
| **Week No.**14 | Packet Filtering Implementations for General-Purpose Computers. |
| **Week No.**15 | System Level Modeling for Hardware/Software Systems and System Level Language Requirements. |
| **Week No.**16 | Presentation of projects and Final Exam. |

## 5- Teaching and Learning Methods

- Lectures
- Tutorials
- Reports & sheets
- Laboratories
- Seminars

**6-Teaching and Learning Methods for Students with Special Needs**

- Lectures
- Tutorials
- Reports & sheets
- Laboratories
- Seminars

The academic advisors of each student, as well as dedicated department TAs monitor the students' progress and solve any problem he/she may encounter.

**7- Student Assessment**

| | | |
|---|---|---|
| **a-Procedures used** | 1-Written Examinations to assess The Intended Learning Outcomes.<br><br>2-Class Activities (Reports, Discussions, ------) to assess The Intellectual Skills. | |
| **b- Schedule:** | Assessment 1<br>Assessment 2<br>Assessment 3<br>Assessment 4 | 7th Week Written Exam<br>12th Week Written Exam<br>Continuous<br>16th Week Final Written Exam |
| **c- Weighing of Assessment** | 7th Week Examination<br>12th Week Examination<br>Final-term Examination<br>Oral Examination<br>Practical Examination<br>Semester Work<br>Total | 30 %<br>20 %<br>40 %<br>00 %<br>00 %<br>10 %<br>100% |

**8- List of References:**

| | |
|---|---|
| **a- Course Notes** | |
| **b- Required Books** (Textbooks) | • Embedded Systems Design: An Introduction to Processes, Tools and Techniques, Arnold S. Berger.<br>• - Object-Oriented Software Engineering: Practical Software Development using UML and Java, Timothy Lethbridge, Robert Laganiere |
| **c- Recommended Books** | |
| **d- Periodicals, Web Sites, etc.** | |

**Course Instructor:**
**Assoc. Prof. Dr. Sherin M. Youssef**

**Program Manager:**
**Prof. Dr. Mohamad AbouEl-Nasr**

**Dean of College of Engineering and Technology of AASTMT**

Name: **Prof. Moustafa Hussein Aly**

Signature:

**Executive Manager of Quality Assurance Center of AASTMT**

Name: **Prof. Aziz Ezzat**

Signature: