sourcemind

# Computer Science & Software Engineering Fundamentals

## Course Description;

This course provides the students with a broad and rudimentary foundation incomputer science, computer architecture and software engineering along with a survey of real-life repercussions of the concepts and common pitfalls in eachcategory. The topics addressed in this course include: a historical review of the evolution of computer architecture, automata, data representations in binary and hexadecimal systems, basic hardware components and interfaces, operating systems, and basic concepts regarding software development lifecycle.

## Course Objectives;

By the end of this course, the students should be able to:

- Demonstrate a clear understanding of the basic concepts of Computer Science: Computer architecture, OS, Data representation, etc.
- Clearly describe Software Development Lifecycle (SDLC), processes and culture

## Prerequisites;

None

## Course Duration;

9 hours

## Preferred Days;

Tuesday

## Required Materials;

A carefully selected set of reading materials and online resources, provided at the end of each class

## Training Topics;

| Session # | Topic |
|---|---|
| 1 | Introduction |
| 2 | Computer Architecture |

| | |
|---|---|
| 3 | Software Engineering |

## Number of Homework, Quizzes and Exams;

| Assignment | Occurrences |
|---|---|
| Homework | 2 |
| Quiz | 3 |

| Session # | Duration | Topics | Session Learning Objective | Materials | Practical Work/Examples | Homework/est. time |
|---|---|---|---|---|---|---|
| 1 | **3 hours** | •History Review<br>•Number Representations Binary,Hex, Float, Two's Complement, Glitches, and pitfalls<br>•Turing Machine, Von NeumannArchitecture<br>•Hardware Components, Chipset,RAM, I/O<br>•Processors, CPU, GPU, Bus, Control unit, ALU, Registers | •Participants will be able to perform arithmetic operations and conversions involving binary, octal and hexadecimal systems.<br>•Participants will be able to explain the risks involving floating point and integer calculations and reproduce scenarios in which buggy phenomena can be observed.<br>•Participants will be able to identify hardware components, explain the purpose and the units of measurement for each and how each contributes to the overall performance of a computer system. | Slides | •Exhibition of physical hardware components<br>•Factorial calculation problem with 32-bit and 64bit signed integers and debugging. | •Reading and research (120 mins)<br>•Homework assignment 1. (180 mins, 25%) |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | **3 hours** | •Cycles, Fetch-Decode-Execute<br>•Process, Virtual Memory, Paging,Threads<br>•Compilation vs. Interpretation, (dis)assembly<br>•Firmware, BIOS, UEFI, MBR, BootSequence<br>•Network, Protocols, Layers, DNS | •Participants will be able to describe the inner workings of running processes.<br>•Participants will be able to explain fundamentals of hardware components, firmware, and drivers.<br>•Participants will be able to identify and describe fundamentals and common practices for organizing communications between hardware components and between computer systems over networks, various protocols, and security.<br>•Participants will be able to construct the model for a rudimentary CPU along with its registers, architecture and instruction set. | Slides | •Architecture Implementation.<br>•Tracing all the inner workings necessary for accomplishing the task of opening a web page. | •Reading and research (120 mins)<br>•Homework assignment 2. (180 mins, 25%)<br>•Quiz1 15% |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | **3 hours** | •Introduction<br>•Software Development Lifecycle<br>•SDLC Models: Agile, Waterfall<br>•Methods, Layers, Tools<br>•Demonstration and Implications | •Participants will be able to demonstrate an understanding of the main concepts of SDLC and relevant terminology along with Waterfall and Scrum workflow models and their phases.<br>•Participants will be able to identify and compose basic SLDC related artifacts; use cases, requirements, etc. | •Slides<br>•Bennatan, E. M. (2000). *On time within budget: Software project management practices and techniques*. New York: Wiley. | •Group activity for specifying requirements of a project.<br>•Group activity about resource management for agile workflows. | •Reading and research (120 mins)<br>•Quiz2 15%<br>•Quiz3 15% |