# INSTITUTION National University of Computer and Emerging Sciences, Karachi Campus

**PROGRAM (S) TO BE**

**EVALUATED**

BS (CS), Fall 2024

# A. Course Description

(Fill out the following table for each course in your computer science curriculum. A filled out form should not be more than 2-3 pages.)

|  |  |
| --- | --- |
| **Course Code** | EE-2003 |
| **Course Title** | Computer Organization and Assembly Language |
| **Credit Hours** | 3 + 1 |
| **Prerequisites by Course(s) and Topics** | ITC, DLD |
| **Assessment Instruments with Weights** (homework, quizzes, midterms, final, programming assignments, lab work, etc.) | Midterms 15 x 2 = 30 marks  Assignments 10 marks  Quiz 10 marks  Final Exam 50 marks |
| **Course Coordinator** | Mr. Shoaib Rauf |
| **URL (if any)** |  |
| **Current Catalog Description** |  |
| **Textbook** (or **Laboratory Manual** for Laboratory Courses) | Assembly Language for Intel Based Computers K.Irvine 6th Edition  MIPS Assembly Language Programming by Ed Jorgensen, Version 1.1.35 April 2018 |
| **Reference Material** | Computer organization and design: the hardware/software interface by David  A. Patterson and John L. Hennessy  Computer Organization & Embedded Systems Hamacher et al. 6th Ed. |
| **Course Goals** | * Programming Methodology of low-level languages * How to access computer hardware directly * Overview of a user-visible architecture (of Intel 80x86 processors) * Intel 80x86 instruction set, assembler directives, macro, etc. * How programs interact with the operating system for various services including memory management and input/output services   How is it possible to interface high-level language and low-level language modules |

|  |  |
| --- | --- |
| **Topics Covered in the Course, with Number of Lectures on Each Topic** (assume 15-week instruction and one- hour lectures) | **Week 1: Introduction to Basic Concepts, Intel 80x86 Processor Architecture** Basic microcomputer architecture  CLO1 |
| **Week 2:** Instruction execution cycle, memory management, input and output systems  CLO1 |
| **Week 3: Assembly Language Fundamentals:** Assembling, Linking and debugging, defining constants and variables, Real and Protected mode Addressing and Programming  **CLO2** |
| **Week 4**: Data transfer and Arithmetic Instructions, Addressing Modes  **CLO2** |
| **Week 5**: Operators and directives, Introduction to control transfer instructions, Arrays and loops, (Addressing modes Contd.) **CLO2** |
| **Week 6**: **FIRST MID TERM EXAMINATION** |
| **Week 7:** Procedures and Stack operations, Runtime stack, PUSH and POP instructions.  **CLO4** |
| **Week 8: Conditional Processing** Boolean and comparison instruction, conditional jumps, conditional loop structures, high-level language constructs  **CLO2** |
| **Week 9: Conditional Processing** (Contd.) Boolean and comparison instruction, conditional jumps, conditional loop instructions, high-level language constructs  **CLO2** |
| **Week 10: Integer Arithmetic**  Shift & Rotate, Multiplication & Division instructions, Extended Addition & Subtraction  **CLO2** |
| **Week 11: SECOND MID TERM EXAM** |
| **Week 12: Advanced Procedures Introduction and Examples:**  Stack Frames, Recursion, INVOKE, ADDR, PROC, PROTO Directives CLO1,CLO2, **CLO4** |
| **Week 13: String and Arrays**  String primitive Instructions, Two dimensional array  **CLO2** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Week 14: Machine Language Translation**  Instruction Formats, encoding an Instruction Set and Modes of Addressing, Translation and Working of an Assembler, Map File and Memory Map **CLO3**  **High level language Interfacing**  Introduction, .model directive, Inline Assembly Code, **Procedures** Linking to an external library  **CLO5**  **Week 15 and 16: MIPS Architecture and Assembly Programming;**  Introduction to MIPS Assembly CISC vs RISC  ILP: Pipelining, Hazards  **CLO 1 and CLO2** | | | |
| **Laboratory Projects/Experiments** | Lab manual available separately | | | |
| **Programming Assignments Done in the Course** | 4 Assignments will be given on the google classroom. | | | |
| **Class Time Spent on** (in credit hours) | Theory | Problem  Analysis | Solution Design | Social and  Ethical Issues |
| 20 | 15 | 15 | 5 |
| **Oral and Written Communications** | Every student is required to submit a project along with its report of not more than 10 pages. | | | |

|  |  |
| --- | --- |
| **PLO** | **Program Learning Outcome (PLO) Statement** |
| 02 | **Problem Analysis:** Ability to identify, formulate, research literature, and analyze  complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences. |
| 05 | **Modern Tool Usage:** Ability to create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and  modeling, to complex engineering activities, with an understanding of the limitations. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLO** | **Course Learning Outcome (CLO)** | **Domain** | **Taxonomy Level** | **PLO** | **Tools** |
| 01 | Illustrate micro-architectures of x86 and RISC processors | Cognitive | 3 | 05 | A1, Q1, M1, F |
| 02 | Create basic assembly code using different type of addressing modes in x86 & RISC ISAs to solve simple-moderate problems | Cognitive | 4 | 02 | A2, M1, F |
| 03 | Apply translation of machine instructions into binary code and visa versa. | Cognitive | 5 | 05 | A2, A3, Q1, M1, M2, F |
| 04 | Illustrate use of stack during a parametrized function/procedure call that uses local variables. | Cognitive | 5 | 05 | A3, M2, F |
| 05 | Justify need to use assembly code along with a high-level language code | Cognitive | 5 | 05 | Q2,A3, M2, F |
| *Tool: A = Assignment, Q = Quiz, M = Midterm, F=Final* | | | | | |