1. *Course number and name*: CSCI 5331 Computer Architecture

2. *Credits and contact hours*: 3 credit, 3 contact

3. *Instructor's or course coordinator's name*: Pradipta De, PhD

4. *Text book, title, author and year*: Essentials of Computer Architecture, Douglas E. Comer
   a. *Other supplemental materials:* None

5. *Specific course information*
   a. *Brief description of the content of the course (Catalog Description)* Digital logic: transistors, circuits, sensors, robotic control; registers and register banks; arithmetic-logic units; data representation: big-endian and little-endian integers; one and twos complement arithmetic; signed and unsigned values; Von-Neumann architecture and bottleneck; instruction sets; RISC and CISC designs; instruction pipelines and stalls; rearranging code; memory and address spaces; physical and virtual memory; interleaving; page tables; memory caches; bus architecture; polling and interrupts; DMA; sensor and device programming; assembly language; optimizations; parallelism; data pipelining. Graduate students will be given extra assignments determined by the instructor that undergraduates will not be required to do.
   b. *Prerequisites*: A minimum grade of "C" in CSCI 3232 or CSCI 3341
   c. *Indicate whether a required, elective, or selected elective course in the program*
      Required course for BS-CS.

6. *Specific goals for the course*
   a. *Specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.*

Course Learning Outcomes	Student Outcomes
Understand Basic Concepts of and the key Milestones in Computer Architecture and Hardware Technology (CPU, bus, motherboard, memory, stored programs, graphics controllers, disk, mice, keyboards, displays, etc.)	1b, 1c, 1d, 1i
Ability to use and understand internal Data Representation (integers and floating point numbers, by IEEE standards, characters in ASCII and UNICODE, as well as strings, arrays and other complex data structures)	1a, 1b, 1c, 1i
Understand Assembly-level organization: instruction fetch, decode, and execution, instruction sets and types, instruction formats, addressing modes, subroutine call and return mechanism (program stack)	1a, 1b, 1c, 1i, 2b

Understand Memory Systems (storage technologies, memory hierarchy, caches, memory management concepts) and their performances	1a, 1c, 1i, 2b
Understand Interfacing, I/O, busses and inter communication and the effects on overall performance	1a, 1c, 1i, 2b
Understand Functional Model of Computer Hardware and Von Neumann Architecture	1a, 1b, 1c, 1i
Understand representation of all components of an executing program and to use debugger to access and manipulate them	1a, 1b, 1c, 1i
Understand contemporary processor architectures (including multi-core) and performance enhancement approaches such as RISC, branch prediction, perfecting, out of order execution, etc.	1a, 1b, 1c, 1i
Understand pipeline concepts and super-scalar architectures	1a, 1b, 1c, 1i
Ability to assess performance of contemporary computers, assembly/configure hardware systems and recommend best value for money configurations	1a, 1b, 1c, 1i, 2a
Ability to program and debug programs using real Assembly Language (MASM) assembly coding standards and secure coding practices, also understanding of instruction cycle, opcodes, mnemonics, effective address, registers, addressing modes, and macros.	1a, 1b, 1c, 1i, 2b
Ability to Interface with a higher level language (HLL) in a real assembly language (e.g. MASM) and impact performance, and understand translation of HLL statements and procedure calls to assembly language	1a, 1b, 1c, 1i, 2b
Understands threads and processes in execution	1a, 1b, 1c, 1i
Understand and use common code optimization techniques and pipeline scheduling	1a, 1b, 1c, 1i
Ability to understand contemporary and advanced computer architectures, research and present a topic	1f

b. Student Outcomes:

- 1a: An ability to apply knowledge of computing and mathematics appropriate to the discipline
- 1b: An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- 1c: An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
- 1d: An ability to function effectively on teams to accomplish a common goal
- 1f: An ability to communicate effectively with a range of audiences
- 1i: An ability to use current techniques, skills, and tools necessary for computing practice
- 2a: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems

in a way that demonstrates comprehension of the tradeoffs involved in design choices

- 2b: An ability to apply design and development principles in the construction of software systems of varying complexity

7. *Brief list of topics to be covered*
   - HiTechnic SuperPro Robots
   - Debug
   - Expression coding
   - Conditionals/loops
   - Visual Studio tracing
   - Gcc/gbd
   - OpenGL
   - Parallel Programming