# EE 5123 --- Computer Architecture (Spring 2015)
# Course Syllabus

**Instructor**:
Lide Duan
http://engineering.utsa.edu/~lideduan/index.html

**Office**: AET 2.376 **Phone**: 210-458-5208
**Email**: lide.duan@utsa.edu
**Office Hours**: 10:00 – 11:30AM (Mondays and Wednesdays) or by appointment
**Website**: All class materials will be handled electronically through Blackboard Learn.
**Class Meeting Time and Location:**
6:00-7:15PM (Mondays and Wednesdays), Mckinney Humanities Building (MH) 3.04.08

**Course Description**:
Computer architecture fundamentals, instruction set architectures, memory and cache architectures, microprocessor pipelining, instruction-level parallelism, data-level parallelism, thread-level parallelism, and request-level parallelism.

**Prerequisite**:
Graduate standing. Non-experience in computer architecture is assumed. However, the course projects require programming skills in high-level languages (e.g. C++, Java, or Python, etc.).

**Textbook**:
Computer Architecture: A Quantitative Approach, 5th Edition. (By John Hennessy and David Patterson)

**Course Topics** (tentative):
- Fundamentals of computer architecture
- Instruction set architectures (ISA)
- Cache and memory hierarchy design
- Pipelining
- Instruction-level parallelism (ILP)
  - Branch predictions, dynamic scheduling, multiple issue and static scheduling, speculations, compile techniques, ILP limitations, etc.
- Data-level parallelism (DLP)
  - Vector architectures, SIMD, GPUs, loop-level parallelism.
- Thread-level parallelism (TLP)
  - Centralized shared-memory, distributed shared-memory, multiprocessor memory coherence, synchronization, consistency, etc.
- Request-level parallelism (RLP)
  - Warehouse-scale computers, cloud computing.

**Grading Policy**:
- Homework assignments: **30%**
  - 6 assignments of 5% each
- Exams: **35%**
  - Test 1 (10%) + Test 2 (10%) + Final (15%)

- Project: **30%**
  - Project 1 (15%) + Project 2 (15%)
- Quizzes: **5%**
  - The instructor will randomly choose a few classes to ask the students to turn in an answer to a very simple question. These are mainly for checking attendance.
- Total: **100%**

**About the Grading**

- The final letter grades will be curved based on the ranks and score gaps.
- After the grade of each assignment/exam/project/quiz is posted, you have up to a week to see me if there is any misgrading or miscalculation. After that, the grade is finalized.

**About the Assignments**

- Tentatively, the 6 assignments will be out in the 3rd, 5th, 8th, 11th, 13th, and 15th week of the semester, and due in about 5 days. No late turn-ins.

**About the Exams**

- All exams are open-books and open-handouts.
- No make-up exams (except for extremely special situation with legitimate proof and under discretion of the instructor).
- Tentatively, Test 1 will be on Wednesday 2/18 (class time), Test 2 will be on Wednesday 4/1 (class time), and Final will be on Monday 5/4 (6pm).

**About the Project**:

- There will be two projects. Both need programming.
- Project 1 will be a disassembler of the MIPS binary code. You will be given a MIPS binary file (in txt file), and need to disassemble it to the MIPS assembly code.
- Project 2 will be a cache performance simulator. You will be given a txt file that has a list of memory addresses, and need to simulate the cache behavior based on specified parameters (e.g. cache size, block size, associativity, replacement policy).
- Tentatively, the two projects will be out in the 3rd / 10th week, and due in the 8th / 15th week, respectively.
- The projects can be implemented in any high-level language (e.g. C, C++, Java, Python, etc.) under any operating system (e.g. Linux, Mac OS, Windows, etc.).
- **Both projects must be completed individually**. There's zero-tolerance on copying source code, either from online or from classmates. Both the copier and copiee will lose all the points for the project.