1. **Department, number and title of course:** Electrical Engineering and Computer Sciences: CS 47C – Completion of Work in Computer Science 61C, Supplemental Machine Structures

2. **Catalog Description:** (1 unit, self-paced, graded) MIPS instruction set simulation. The assembly and linking process. Caches and virtual memory. Pipelined computer organization. Students with sufficient partial credit in CS 61C may, with the consent of the instructor, complete the credit in this course.

3. **Prerequisites:** Experience with assembly language programming that includes writing an interrupt handler, plus CS 9C or equivalent, in addition to the consent of the instructor. Students will not receive credit for CS 61C taken after CS 47C, or for CS 47C taken after CS 61C.

4. **Textbooks and/or other required material:**

5. **Course objectives:** CS47C brings students through a series of abstractions from high-level programming through machine architecture to logic design. The C programming language, MIPS assembly language, and schematic diagrams are used to introduce the abstractions. The course closely follows the Patterson and Hennessy textbook, supplemented by material on the C programming language and notes on the design of synchronous digital systems.

6. **Topics Covered:**
   Course activities include programming assignments and quizzes; quizzes focus on low-level language details or programming techniques, while programming assignments are broader in scope. One of the programs is a substantial project comprising several hundred lines of code. The list of programming assignments appears below. The topics of assembly and linking are covered in a quiz. Programs are coded in the C programming language.
   - MIPS interpreter: Students write a simulator for a subset of the MIPS machine language.
   - Cache exercises: Students add cache simulation, and infer cache characteristics from access time data.
   - Virtual memory exercises: These exercises are drawn from the Patterson and Hennessy textbook. They deal with the mechanisms of address translation and page table use.
   - Pipeline exercises: These exercises are drawn from the Patterson and Hennessy textbook. They involve identifying and optimizing pipeline hazards in program code.

7. **Class/laboratory schedule:** Self-paced.

8. **Contribution of course meeting the professional component:** The course contributes to engineering topics. It is approximately 2/3 science and 1/3 design.

9. **Relationship of course to program objectives:** Biweekly independent projects reinforce students’ programming skills while reinforcing their understanding of machine architecture and implementation concepts. Sample projects include: a machine language interpreter written in C; a cache simulator written in C; and a CPU hardware design and implementation.