# http://www.ric.edu/webcommunications/images/SealWithText_Small_Black.pngUNDERGRADUATE CURRICULUM COMMITTEE (UCC) PROPOSAL FORM

## Cover page scroll over blue text to see further important [instructions](#instructions): [if not working select “COMMents on rollover” in your Word preferences under view] please read these.

**N.B. Please do not use highlight to select choices within a category but simply delete the options that do not apply to your proposal (e.g. in A.2 if this is a course revision proposal, just delete the creation and deletion options and the various program ones, so it reads “course revision”) Do not ever delete any of the numbered categories—if they do not apply leave them blank. ALL numbered categories in section (A) must be completed. If there are no resources impacted it is okay to put “none” in A. 7**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A.1. [Course or program](#Proposal) | **CSCI 312 - Computer Organization and Architecture I (delete)****CSCI 313 Computer Organization and Architecture (REVISE)** | | | |  |
| [Replacing](#Ifapplicable) |  | | | |
| A. 1b. Academic unit | **Faculty of Arts and Sciences** | | | |  |
| A.2. [Proposal type](#type) | **Course: revision** **and deletion** | | | |  |
| A.3. [Originator](#Originator) | **Robert Ravenscroft, Namita Sarawagi** | [Home department](#home_dept) | **COMPUTER SCIENCE AND INFORMATION SYSTEMS** | | |
| A.4. [Context and Rationale](#Rationale)  Note: Must include this additional information for all [new programs](#type) | **Currently there are two courses CSCI 312 (4 credits) and 313 (3 credits), titled Computer Organization and Architecture I and II both required in the BA and BS Computer Science majors. The two courses are being combined into a single 4 credit course.**  **CSCI 313 is being updated to include necessary components from CSCI 312.**  **These changes also reflect the updated ACM Curricula Guidelines** <https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf>  **The credits for CSCI 313 are increasing from 3 to 4 and CSCI 312 is being deleted.**  **The title of this updated CSCI 313 course will be Computer Organization and Architecture, and the description and prerequisites are being updated to reflect this change, with CSCI 212/CSCI 212W replacing CSCI 211/221.**  **In most other schools there is only one course in Computer Organization and Architecture, so with this change we will be consistent with other programs.** | | | | |
| A.5. [Student impact](#student_impact) | **Will help students take this course and the next course, CSCI 435, for which CSCI 313 is a pre-requisite, earlier, thus not delaying their graduation.** | | | | |
| A.6. [Impact on other programs](#impact) | **None.** | | | | |
| A.7. [Resource impact](#Resource) | [*Faculty PT & FT*](#faculty): | **None** | | | |
| [*Library*:](#library) | **None** | | | |
| [*Technology*](#technology) | **None** | | | |
| [*Facilities*](#facilities): | **None** | | | |
| A.8. [Semester effective](#Semester_effective) | **Fall 2021** | A.9. [Rationale if sooner than next Fall](#Semester_effective) | |  | |
| A.10. INSTRUCTIONS FOR CATALOG COPY: This single file copy must include ALL relevant pages from the college catalog, and show how the catalog will be revised. (1) Go to the “Forms and Information” page on the UCC website. Scroll down until you see the Word files for the current catalog. (2) Download ALL catalog sections relevant for this proposal, including course descriptions and/or other affected programs. (3) Place ALL relevant catalog copy into a single file. Put page breaks between sections and **delete any catalog pages not relevant for this proposal**. (4) Using the track changes function, revise the catalog pages to demonstrate what the information should look like in next year’s catalog. (5) Check the revised catalog pages against the proposal form, especially making sure that program totals are correct if adding/deleting course credits. If new copy, indicate where it should go in the catalog. If making related proposals a single catalog copy that includes all is preferred. Send catalog copy as a separate single Word file along with this form. | | | | | |

B. [NEW OR REVISED COURSES](#delete_if)  **Delete section B if the proposal does not include a new or revised course. As in section A. do not highlight but simply delete suggested options not being used. Always fill in b. 1 and B. 3 for context.**

|  | Old ([for revisions only](#Revisions)) ONLY include information that is being revised, otherwise leave blank. | New Examples are provided within some of the boxes for guidance, delete just the examples that do not apply. |
| --- | --- | --- |
| B.1. [Course prefix and number](#cours_title) | **CSCI 313** | **CSCI 313** |
| B.2. Cross listing number if any |  |  |
| B.3. [Course title](#title) | Computer Organization and Architecture II | Computer Organization and Architecture |
| B.4. [Course description](#description) | A continuation of CSCI 312. Topics include the central processing unit, memory access, input/output, and floating point operations. | Students investigate combinational and sequential circuits. System architecture including the central processing unit, memory, input/output. MIPS assembly language programming. Input/output and interrupt programming. System performance enhancements including caching and parallelism. |
| B.5. [Prerequisite(s)](#prereqs) | CSCI 312 and either CSCI 211 or CSCI 221 | CSCI 209 or CSCI 312; and CSCI 212 or CSCI 212W. |
| B.6. [Offered](#Offered) |  |  |
| B.7. [Contact hours](#contacthours) | **3** | **4** |
| B.8. [Credit hours](#credits) | **3** | **4** |
| B.9. [Justify differences if any](#differences) |  | |
| B.10. [Grading system](#grading) |  |  |
| B.11. [Instructional methods](#instr_methods) |  |  |
| B.11.a [Delivery Method](#instr_methods) |  |  |
| B.12.[Categories](#required) | **Required for major** | **Required for major** |
| B.13. Is this an Honors course? | **NO** | **NO** |
| B.14. [General Education](#ge)  N.B. Connections must include at least 50% Standard Classroom instruction. | **NO** | **NO** |
| B.15. [How will student performance be evaluated?](#performance) | **Exams | Class Work | Quizzes | Projects |** | **Exams | Class Work | Quizzes | Projects |** |
| B.16 [Recommended class-size](#class_size" \o "Check appendix XVIII in the UCC Manual for Best Practices) | **24** | **24** |
| B.17. [Redundancy statement](#competing) |  |  |
| B. 18. Other changes, if any |  | |

| B.19**.** [**Course learning outcomes**](#outcomes)**: List each one in a separate row** | [**Professional Org.Standard(s)**](#standards)**, if relevant** | [**How will each outcome be measured**](#measured)**?** |
| --- | --- | --- |
| Design combinational circuits. |  | Projects, quizzes, exams. |
| Explain the behavior of sequential circuits. |  | Projects, quizzes, exams. |
| Describe the architecture of a computer. |  | Projects, quizzes, exams. |
| Describe the architecture of a CPU. |  | Projects, quizzes, exams. |
| Describe the architecture of computer memory. |  | Projects, quizzes, exams. |
| Describe the architecture of computer i/o. |  | Projects, quizzes, exams. |
| Describe the structure of an assembly language program |  | Projects, quizzes, exams. |
| Describe and code expressions and data storage in assembly. |  | Projects, quizzes, exams. |
| Describe and code functions in assembly. |  | Projects, quizzes, exams. |
| Describe and code loops and decisions in assembly. |  | Projects, quizzes, exams. |
| Describe and code arrays and structures in assembly. |  | Projects, quizzes, exams. |
| Explain caching. |  | Projects, quizzes, exams. |
| Explain sources of parallelism in a CPU design. |  | Projects, quizzes, exams. |
| Explain i/o programming with polling and interrupts. |  |  |
|  |  |  |

| B.20. [**Topical outline**](#outline)**: DO NOT INSERT WHOLE SYLLABUS, JUST A TWO-TIER TOPIC OUTLINE. Proposals that ignore this request will be returned for revision.** |
| --- |
| 1. Combinational Circuits (1.5 weeks)    1. Logic Gates    2. Logic circuit design    3. Multiplexers    4. Addition Circuits    5. Comparator Circuits 2. Sequential Circuits (2 weeks)    1. Flip Flops    2. Latches    3. Clocks    4. Counters    5. Multiplication Circuits    6. ALU Design 3. History of Computing (1/2 week)    1. Historical Computation Devices    2. Mechanical Computers    3. Vacuum Tube Computers    4. Transistor Computers    5. Integrated Circuit Computers 4. System Architecture (1/2 week)    1. Parts of a computer: CPU, Memory, I/O    2. Von Neumann Architecture    3. Harvard Architecture    4. Microcontrollers and embedded devices    5. System bus and bus contention    6. MIPS and Intel architectures    7. Flynn’s Taxonomy 5. CPU Architecture (1/2 week)    1. ALU    2. Registers    3. Control Logic    4. Fetch-Execute Cycle    5. Data Path    6. Machine language 6. Memory Architecture (1/2 week)    1. Random access, volatile, non-volatile    2. Dynamic RAM, static RAM    3. Types of ROM and Flash memory    4. Asynchronous memory    5. Synchronous memory 7. I/O Architecture (1/2 week)    1. I/O controllers    2. Data and control registers    3. Serial and parallel interfaces, USB    4. Analog versus digital data    5. Memory mapped versus I/O bus 8. MIPS – Assembly Basics (1 week)    1. Assembly language    2. Assembler    3. MIPS registers    4. Assembler directives    5. Pseudo instructions and macros    6. Basic data manipulation 9. MIPS Assembly – Expressions and Assignment (1 week)    1. Data Allocation and global variables    2. Register allocation of local data    3. Arithmetic and logic operation    4. Memory data transfer operations    5. Expression evaluation    6. Register reuse 10. MIPS – Control Structures (1 week)     1. Conditional and unconditional branching     2. MIPS set on condition instructions     3. Logical expressions     4. Decision statement     5. Conditional and counted loops 11. MIPS – Structured Data (0.5 week)     1. Arrays, allocation and access     2. Loop counter and arrays     3. Structures, allocation and access     4. Memory alignment 12. MIPS – Functions (0.5 week)     1. Function calls and returns     2. Parameter passing by register     3. Overview of run-time stack 13. Instruction Set Architecture (0.5 week)     1. Expanding opcodes     2. Prefix opcodes     3. Addressing modes for operands     4. RISC versus CISC 14. Performance Enhancements (1 week)     1. Caching and instruction pre-fetch     2. Pipeline     3. Branch prediction     4. Superscalar architecture     5. Processor threading     6. Instruction reordering     7. Register renaming 15. I/O Programming (1 week)     1. Polling versus interrupts     2. Interrupt handlers     3. Vectored interrupts     4. Priority interrupts     5. Shadow registers     6. Bit twiddling 16. Buses (0.5 week)     1. Parallel versus serial     2. Data skew     3. Attenuation     4. Radio frequency interference     5. Bridges 17. Testing and evaluation (1 week) |

## D. Signatures

* Changes that affect General Education in any way MUST be approved by ALL Deans and COGE Chair.
* Changes that directly impact more than one department/program MUST have the signatures of all relevant department chairs, program directors, and their relevant dean (e.g. when creating/revising a program using courses from other departments/programs). Check UCC manual 4.2 for further guidelines on whether the signatures need to be approval or acknowledgement.
* Proposals that do not have appropriate approval signatures will not be considered.
* Type in name of person signing and their position/affiliation.
* Send electronic files of this proposal and accompanying catalog copy to [curriculum@ric.edu](mailto:curriculum@ric.edu) and a printed signature copy of this whole form to the current Chair of UCC. Check UCC website for due dates.

##### D.1. Approvals: required from programs/departments/deans who originate the proposal. THESE may include multiple departments, e.g., for joint/interdisciplinary proposals.

| Name | Position/affiliation | [Signature](#_Signature" \o "Insert electronic signature, if available, in this column) | Date |
| --- | --- | --- | --- |
| Lisa Bain | Chair of Computer Science and Information Systems | \*approved via e-mail | 12/03/2020 |
| Earl Simson | Dean of Arts and Sciences | **Earl Simson** | 12/03/2020 |

##### D.2. [Acknowledgements](#acknowledge): REQUIRED from OTHER PROGRAMS/DEPARTMENTS (and their relevant deans if not already included above) that are IMPACTED BY THE PROPOSAL. SIGNATURE DOES NOT INDICATE APPROVAL, ONLY AWARENESS THAT THE PROPOSAL IS BEING SUBMITTED. CONCERNS SHOULD BE BROUGHT TO THE UCC COMMITTEE MEETING FOR DISCUSSION; all faculty are welcome to attend.

| Name | Position/affiliation | [Signature](#Signature_2) | Date |
| --- | --- | --- | --- |
|  |  |  |  |