# 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): please read.

**N.B. DO NOT USE HIGHLIGHT, please DELETE THE WORDS THAT DO NOT APPLY TO YOUR PROPOSAL**

**ALL numbers in section (A) need to be completed, including the impact ones.**

|  |  |  |
| --- | --- | --- |
| A.1. [Course or program](#Proposal) | **CIS 301- Introduction to Computer Programming in Business**  |  |
| [Replacing](#Ifapplicable)  | **CIS 255, CIS 256 and CIS 257** |
| A.2. [Proposal type](#type) | **Course: creation**  |
| A.3. [Originator](#Originator) | **Dr. Lisa Bain** | [Home department](#home_dept) | **Accounting and CIS** |
| A.4. [Context and Rationale](#Rationale)  | CIS majors currently are required to choose one programming language class from a list of five courses:CIS 255 Introduction to Java in Business CIS 256 Introduction to COBOL Programming CIS 257 Introduction to Visual Basic in Business CIS 355 Advanced Business Applications in Java CIS 357 Advanced Business Applications in Visual BasicWe are proposing that the three introductory courses be replaced by one introductory programming class which does not list a specific programming language.  The CIS program does not focus on computer programming skills like the Computer Science program. It is important that CIS majors understand the main concepts of computer programming and have the opportunity to write/test code. Most CIS majors struggle with computer programming, hence not being a Computer Science major. Some Information Systems (IS) programs have removed computer programming from their curriculum, making it an elective. The most current model curriculum (IS 2010) for Information Systems (IS) Programs removed programming as a recommended required course.  However, the CIS faculty strongly feel that a basic foundation in computer programming is very important for CIS majors. In Fall 2014, the CIS major reduced the computer programming requirement from TWO courses to ONE but kept it as a requirement. The intention is not to make students experts in one language but provide coverage of the environment from which to build upon, if interested. In addition, many CIS graduates work for organizations in the role of a systems analyst. This role needs to understand computer programming but does not perform coding as a job requirement.  Students taking the new courses will have the foundation necessary to pursue language-specific computer programming courses, if interested. This includes courses in the Computer Science department. In fact, CIS majors have taken these courses with the encouragement of CIS faculty. These courses can then be used as one of the CIS electives. It is not common for CIS majors to do this but the opportunity is there, providing flexibility in our program.  The CIS faculty will use Python in Fall 18 as the primary language for the course with plans to introduce short lessons/mini labs in other common languages (e.g. JavaScript and Ruby) as time allows. This will help broaden the overall understanding of the computer programming environment for CIS majors. Basic programming concepts, like variables and constructs, would be transferable to other modern languages. The primary language would only change after careful consideration, consensus, and planning by the CIS faculty. |
| A.5. [Student impact](#student_impact) | No negative impacts are foreseen as students are already required to take one of the CIS programming courses. 1. CIS majors that struggle with computer programming will have the opportunity to try new languages that are designed for beginners/novices.
2. CIS majors will still be exposed to the concept of computer programming and its overall importance in an IT department.

CIS majors that find computer programming interesting will have the foundation to pursue additional programming courses and/or programming languages. |
| A.6. [Impact on other programs](#impact)  | **None.** |
| A.7. [Resource impact](#Resource) | [*Faculty PT & FT*](#faculty):  | **None. Faculty already teaching CIS 255, 256 and 257.** |
| [*Library*:](#library) | **None** |
| [*Technology*](#technology) | **None** |
| [*Facilities*](#facilities): | **None. Existing facilities are already being used to teach CIS255, 256 and 257.** |
| A.8. [Semester effective](#Semester_effective) | **Fall 2018** | A.9. [Rationale if sooner than next Fall](#Semester_effective) |  |

B. [NEW OR REVISED COURSES](#delete_if)  **DO NOT use highlight. Delete this whole page if the proposal does not include a new or revised course.**

|  | Old ([for revisions only](#Revisions))Only include information that is being revised, otherwise leave blank (delete provided examples that do not apply) | NewExamples are provided for guidance, delete the ones that do not apply |
| --- | --- | --- |
| B.1. [Course prefix and number](#cours_title)  |  | **CIS 301** |
| B.2. Cross listing number if any |  |  |
| B.3. [Course title](#title)  |  | **Introduction to Computer Programming in Business** |
| B.4. [Course description](#description)  |  | **Introductory course using an object-oriented programming language to solve business problems. Topics include: algorithm concepts and development; object-oriented programming methodologies; graphical interface design and event based programming.** |
| B.5. [Prerequisite(s)](#prereqs) |  | **CIS 252 or CIS 251** |
| B.6. [Offered](#Offered) |  | **Fall | Spring**  |
| B.7. [Contact hours](#contacthours)  |  | **4** |
| B.8. [Credit hours](#credits) |  | **4** |
| B.9. [Justify differences if any](#differences) |  |
| B.10. [Grading system](#grading)  |  | **Letter grade**  |
| B.11. [Instructional methods](#instr_methods) |  | **Laboratory | Lecture**  |
| B.12.[Categories](#required) |  | **Required for CIS major only** |
| B.13. Is this an Honors course? |  | **NO** |
| B.14. [General Education](#ge)N.B. Connections must include at least 50% Standard Classroom instruction. |  | **NO**  |
| B.15. [How will student performance be evaluated?](#performance) |  | **Attendance | Class participation | Exams | Presentations |** **Class Work | Quizzes |****Projects**  |
| B.16. [Redundancy statement](#competing) |  | **The closest course offered at RIC would be CSCI 201 Computer Programming and Design.** **However, CSCI 201 specifically utilizes the Java programming language & does not focus exclusively on programming within the business domain.** |
| B. 17. Other changes, if any |  |

| B.18**.** [**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)**?** |
| --- | --- | --- |
| **Understand basic computing, object-oriented, and programming concepts -- independent of language** **implementation:**  |  | **There will be multiple in class exams which will contain objective questions and programming problems.** |
| **Ability to design algorithms for given a problem situation and** **implement using programming language.**   |  | **Students will complete a significant number of programming assignments of increasing complexity. These programming assignments will require that students demonstrate the** **ability to design, code and test programs.**  |
| **Ability to test and debug programs.**  |  | **Students will be required to find and fix syntax, logic & runtime errors.** **Students will apply techniques for preventing errors.** |
| **Ability to apply standard documentation practices.**  |  | **Students will be required to internal and/or external documentation for all assignments.** |

| B.19. [**Topical outline**](#outline)**: Do NOT insert whole syllabus, we just need a two-tier outline** |
| --- |
| **Topic** |
| 1. Introduction to Computers and Programming:
2. Hardware and software
3. How computers store data
4. How a program works
5. Designing a program

2) Variables, Data Types, Constants, Arithmetic Expressions1. Variables
2. Data types
3. Performing calculations
 |
| 1. Decision Structures
2. Relational operators
3. The if Statement
4. The if-else Statement
5. Switch statement
6. Logical operators
 |
| 1. Repetition Structures
2. Top testing loops
3. Bottom testing loops
4. Counted loops (For)
5. Conditional loops (While/Do)
 |
| 1. Functions/Procedures/Methods
2. Defining and calling functions
3. Designing a program to use functions
4. Local vs global variables
5. Passing arguments to functions
 |
| 6) Files and Exception Handling1. Introduction to file input and output
2. Processing files
3. Exceptions and exception handling
 |
| 7) Classes and Object-Oriented Programming1. Procedural and object-oriented programming
2. Classes design
3. Inheritance
4. Polymorphism
5. Techniques for designing classes
 |

## D. Signatures

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

| Name | Position/affiliation | [Signature](#_Signature" \o "Insert electronic signature, if available, in this column) | Date |
| --- | --- | --- | --- |
| Lisa Bain | Chair of Accounting & CIS |  |  |
| Jeff Mello | Dean of School of Business |  | Tab to add rows |

##### D.2. [Acknowledgements](#acknowledge): REQUIRED from OTHER PROGRAMS/DEPARTMENTS 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

| Name | Position/affiliation | [Signature](#Signature_2) | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Tab to add rows |