# 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. DO NOT USE HIGHLIGHT, where choices are given within categories, please DELETE those THAT DO NOT APPLY TO YOUR PROPOSAL. Do not delete numbered categories.**

**ALL numbers in section (A) to be completed, including the impact ones (#5-7), put “none” if that is the case.**

|  |  |  |
| --- | --- | --- |
| A.1. [Course or program](#Proposal) | **CSCI 309 OBJECT-ORIENTED DESIGN** |  |
| [Replacing](#Ifapplicable)  |  |
| A.2. [Proposal type](#type) | **Course: creation**  |
| A.3. [Originator](#Originator) | **NAMITA SARAWAGI** | [Home department](#home_dept) | **MATHEMATICS AND COMPUTER SCIENCE** |
| A.4. [Context and Rationale](#Rationale)  | This new course, titled Object-Oriented Design, is being proposed as a required course in both the BA and BS Computer Science programs. The objective of this course is to instill in students a proficient understanding of the practice of object-oriented design and programming.The course places an emphasis on the concepts and patterns of object-oriented design and on how to apply those concepts when programming, rather than on the syntax of a specific language. After taking this course, students will have enhanced their competence in design and programming skills and will be better prepared for upper level courses such as CSCI 401 – Software Engineering. This course has been piloted as a Topics course (CSCI 350) since Fall 2018 and has a very positive response and feedback from students. The class size is capped at 24 as this course is taught in a computer lab. |
| A.5. [Student impact](#student_impact) | After taking this course, students will have enhanced their competence in design and programming skills and will be better prepared for upper level courses such as CSCI 401 – Software Engineering. |
| A.6. [Impact on other programs](#impact)  | **CIS** – In the restricted electives for the BS in CIS students can taketwo additional courses in computer information systems or **computer science** at the 300-level or above   |
| A.7. [Resource impact](#Resource) | [*Faculty PT & FT*](#faculty):  | **Existing faculty can teach this course** |
| [*Library*:](#library) | **No additional resources needed**  |
| [*Technology*](#technology) | **Will teach in the existing computer labs using existing software** |
| [*Facilities*](#facilities): | **Existing labs** |
| A.8. [Semester effective](#Semester_effective) | **Fall 2020** | 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 acceptable. Send as a separate single file along with this form. |

B. [NEW OR REVISED COURSES](#delete_if)  **DO NOT use highlight. Do not delete numbered categories, just leave blank if they do not apply. Delete this whole page if the proposal does not include a new or revised course. 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.  | NewExamples 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 309** |
| B.2. Cross listing number if any |  |  |
| B.3. [Course title](#title)  |  | **OBJECT-ORIENTED DESIGN** |
| B.4. [Course description](#description)  |  | **Students will learn fundamental concepts, techniques and principles in object-oriented analysis and design. Topics include the object-oriented design process, interfaces, inheritance, polymorphism, graphical user interfaces, and design patterns.** |
| B.5. [Prerequisite(s)](#prereqs) |  | CSCI 211 or CSCI 201 |
| 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) |  | **Lecture**  |
| B.12.[Categories](#required) |  | **Required for major** |
| 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) |  | **Exams | Class Work | Quizzes | Projects | Team Project** |
| B.16 [Recommended class-size](#class_size" \o "Check appendix XVIII in the UCC Manual for Best Practices) |  | **24** |
| B.17. [Redundancy statement](#competing) |  |  |
| B. 18. 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)**?** |
| --- | --- | --- |
| Read, understand, and write polymorphic code for both interface and inheritance polymorphism |  | Projects, homework and exam/quiz questions. |
| Design to an interface |  | Projects, homework and exam/quiz questions. |
| Design a group of collaborating objects to implement a given set of requirements |  | Projects, homework and exam/quiz questions. |
| Assign the responsibilities for the actions implied by the requirements to appropriate objects |  | Projects, homework and exam/quiz questions. |
| Partition these responsibilities into appropriately named methods |  | Projects, homework and exam/quiz questions. |
| Write definitions for the classes that can be used to generate these objects |  | Projects, homework and exam/quiz questions. |
| Deduce the data necessary to implement the requirements, and partition that data among the objects |  | Projects, homework and exam/quiz questions. |
| Define appropriate instance variables to store these data |  | Projects, homework and exam/quiz questions. |
| Create super-classes in order to share common code |  | Projects, homework and exam/quiz questions. |
| Arrange the concrete and abstract classes in an appropriate hierarchy |  | Projects, homework and exam/quiz questions. |
| Read and use UML appropriately to communicate their designs |  | Projects, homework and exam/quiz questions. |
| Explain the history and purpose of object-oriented design patterns, and recognize, explain a few specific design patterns.  |  | Projects, homework and exam/quiz questions. |

| B.19. [**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. **Fundamentals of Object-Oriented Programming**
* Objects and Classes
* Primitive vs. reference types
* Inheritance
* Overriding, dynamic method lookup
* Polymorphism via Inheritance
* Generic Classes - ArrayList
1. Abstract Classes and Interfaces
* Abstract Classes
* Interfaces – Comparable, Iterator
* Design an Interface Type
* Polymorphism via Interfaces
1. Object Oriented Design
* Static Class Members – static variables and methods
* Class Relationships: Association, Inheritance, Dependency, Aggregation,
* UML Diagrams
1. Class Design Principles
* The Importance of Encapsulation
* Refactoring Code
* Analyzing the Quality of the Interface of a Class - The Five C’s
1. Object-Oriented Design Process
* From Problem to Code – Case Study
* Identifying Classes and Responsibilities
* Relationships Between Classes
* Use Cases
* CRC Cards
* UML Class Diagrams
* Sequence Diagrams
* State Diagrams
1. Graphical User Interfaces
* The Icon Interface Type
* Frames and User Interface Components
* User Interface Actions
* Timers
* Drawing Shapes
* The Hierarchy of Swing Components
1. Exceptions
* Exception Handling
* Uncaught and Caught Exceptions
* Exception Propagation
* The Hierarchy of Exception Classes
* I/O Exceptions
1. Design Patterns – Introduction
* Types of Design Patterns
* Specific Design Patterns – such as:

The Adapter Pattern, The Singleton Pattern, The Visitor Pattern* MVC Architecture

Testing and Review     |
|  |

## 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 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. 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 |
| --- | --- | --- | --- |
| Stephanie Costa | Chair, Mathematics and Computer Science |  |  |
| Earl Simson | Dean, Arts and Sciences |  |  |

##### 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 |
| --- | --- | --- | --- |
| Lisa Bain | Chair, Accounting and Computer Information Systems |  |  |
| Jeffrey Mello | Dean, School of Business |  |  |