LSC Use Only Proposal No:	UWUCC Use Only Proposal No: 11-124n
LSC Action	

all information. Use a separate cove sheet for each course proposal a

Course Number and/or Title Chan

ourse is also propos as a Liberal Stud

urse)

3. Designations, as appropriate

□ Honors College Course □ Other: (e.g. Women's Studies, Pan African)

□ New Degree Program □ New Minor Program □ Liberal Studies Requirement Changes □ Other

Current program name:

roposed program name, if chang

	Signature	Date
urriculum Commi		
hairperson(s)		
College Dean		
Director of Honors College (as needed)		
UCC		

Course Revision –Computer Science Curriculum

Part II. Description of Curriculum Change

1. New Syllabus of Record

I. Course Description

COSC 210 Object Oriented and GUI Programming 3c-0l-3cr

Prerequisite: COSC 108 or COSC 110

An in-depth introduction to the Object Oriented Programming (OOP) paradigm, including encapsulation

inheritance, and polymorphism. The focus will be on designing, implementing, and using objects. This course will also include an introduction to Graphical User Interface (GUI) design and programming.

II. Course Outcomes

Upon successful completion of the course, the student will be able to:

1. Demonstrate an understanding of the fundamental concepts of the OOP paradigm including encapsulation, inheritance, polymorphism, interfaces, and overloading/overriding.
2. Demonstrate an understanding of the fundamental concepts of GUI programming.
3. Design and implement small applications (e.g., involve around a dozen source files) using an object

oriented programming language. A few applications will use a GUI interface.

3. Control Statements 3 hours

- Algorithms
- if, then, while, switch, break, continue, and for
- Expressions and assignments re-visited; operators, precedence.
- Data type handling, object vs. primitive, identity vs. equality
- Exception handling
- Programming conventions part 3

4. More on Object Definitions 3 hours

- Static fields and methods
- Overloading
- Packages (or namespaces) and encapsulation revisited
- Intro to API libraries/packages
- Parameter passing
- Argument promotion and casting

5. Arrays 3 hours

- Declaring and creating arrays
- Initializing an array
- Arrays of Objects
- For each statement
- Multidimensional arrays
- [illegible]

- Polymorphism behavior using arrays.
- Final attribute on methods and classes
- Abstract classes vs. concrete classes

10. Interfaces 3 hours

- Definition of an interface, a contract
- Implementation vs. use
- Programming to an interface
- Polymorphism behavior using arrays.
- Final attribute on methods and classes
- UML class diagrams for interfaces

- Revisit IO
- Collections APIs
- List, Iterator, and Map interfaces

IV. Evaluation Method

Grade distribution:

Quizzes	10%
Mid Term (1 hour 15 min)	15%
Final Exam (Cumulative)	20%
Around 8 Programming Projects	50%
Participation	5 %

Grade Scale:

90 – 100%	A
80 – 89%	B
70 – 79%	C
60 – 69%	D
< 60%	F

Attendance Policy:

Attendance is crucial to success in this course. To encourage class attendance, the following policy will be used: Attendance will be taken at every class. For each unexcused absence, starting with the fourth, 2% will be deducted from the overall class grade. Generally, excused absences involve illness with a doctor's

Course Revision –Computer Science Curriculum

1. Anderson, Julie and Franceschi, Herve'. *Java 6 Illuminated*, 2nd ed., Jones and Bartlett, Sudbury, MA 2008.

NJ. 2004.

3. Friedman, Frank L. and Koffman, Elliot B. Problem Solving, Abstraction, and Design Using C++, 6th ed.

Course Revision –Computer Science Curriculum

2. Summary of Proposed Revisions

Prerequisites have been changed to COSC 108 or COSC 110.

The course outcomes are essentially the same, but have been enhanced to provide clarification and use measures targeting higher levels in Bloom's taxonomy. Two outcomes were added in relational to adapt to current trends in the industry. The additional outcomes relate to industry programming practices (outcome

Course Revision –Computer Science Curriculum

4. The Old Syllabus of Record

I. Course Description

COSC 210 Object Oriented and GUI Programming

Prerequisite: COSC110

An in-depth introduction to the Object Oriented Programming (OOP) paradigm. The focus will be on designing, implementing, and using objects. We will cover function and operator overloading, templates, inheritance, and polymorphism. This course will also include an introduction to Graphical User Interface (GUI) design and programming.

II. Course Objectives

The student will:

1. Learn the fundamental concepts of the OOP paradigm.
2. Implement object definitions.
3. Incorporate objects and arrays of objects in application programming

3. Operator overloading
4. Friend functions

E. Using Objects 4 hours

1. Scoping
2. Static and dynamic objects
3. Arrays of objects

F. Object Design 3 hours

1. Goals of Object Oriented Software

G. Templates 4 hours

[redacted]

2. Simple sorting techniques (exchange, insertion, selection)
3. Template object definitions

H. Inheritance 4 hours

1. Derived types
2. Virtual functions
3. Declarations
4. Single and multiple inheritance
5. Public vs. private inheritance
6. Virtual Derivations

I. Run-Time Type Identification 2 hours

1. Polymorphism

J. Exception Handling and Debugging 3 hours

1. Exception handling functions
2. Expected and unexpected exceptions
3. Using a debugger

[redacted]

[redacted]

1. Goals of GUI software

L. Implementing GUIs using Object Oriented Programming 5 hours

1. Concepts and terminology

Project 4. Develop a hierarchical set of classes such as quadrilaterals. This project will include inheritance, polymorphism, protected attributes and methods and overloading.

Project 5. Develop a Dialog Based Graphical User Interface application. This project will introduce the concepts of GUI design and development. The project will also include the use of a large library of predefined classes.

V. Textbook(s)

Deitel, H.M. and Deitel, P.J.
C++ How to Program. Second Edition.
Prentice Hall PTR, Upper Saddle River, NJ. 1998.

Deitel, H.M., Deitel, P.J., Nieto, T.R., and Strassberger, E.T.

Part II. Letters of Support

Not Applicable. This is an internal change to the Computer Science program. Affected programs are all tracks in the Computer Science major, the Computer Science minor, and the Information Assurance Minor
.