

LAST REVIEW:
2009-2010

NEXT REVIEW: 2014-15
2014-2015

STATUS: A
A

COURSE TITLE: Object-Oriented Analysis and Design

COMMON COURSE NUMBER: COP 2361C

CREDIT HOURS: 3

CONTACT HOUR BREAKDOWN
(per 16 week term)

CLOCK HOURS:

Lecture: **40** Lab: **8**
Clinic: Other:

PREREQUISITE(S): CIS 2321C and COP 1335C (each with a grade of C or higher)

COREQUISITE(S): None

PRE/COREQUISITE(S): None

COURSE DESCRIPTION:

This course focuses on the object-oriented software development process, including object-oriented methodologies and workflows. Students will be able to determine the Use Cases and Domain Model of the problem domain. Create a system design supporting functional requirements. Create a system architecture supporting the nonfunctional requirements and development constraints.

General Education Requirements – Associate of Arts Degree (AA), meets Area(s): Area
General Education Requirements – Associate in Science Degree (AS), meets Area(s): Area
General Education Requirements – Associate in Applied Science Degree (AAS), meets Area(s): Area

UNIT TITLES

1. Software Development Process
2. Object-Oriented Methodologies
3. Use Case Diagrams
4. Problem Domain Model - Class Diagrams
5. Design Model – Collaboration and Sequence Diagrams
6. Fundamental Architecture Concepts
7. Domain Model Refining
8. Design Patterns
9. Statechart Diagrams

EVALUATION:

Please provide a brief description (250 characters maximum) that details how students will be evaluated on the course outcomes.

Evaluation instruments will include written and/or skills-based examination and individual in-class and/or take-home assignments. Evaluation methods may also include group in-class and/or take-home assignments.

Common Course Number: COP 2361C

UNITS

Unit 1 Software Development Process

General Outcome:

1.0 The student shall:

Understand the concept of software development using Object-Oriented methodologies. Comprehend how the different Object-Oriented Software Development workflows support the Analysis, Design, Implementation, and Maintenance phases of software development.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 1.1 Describe the Object-Oriented Software Development (OOSD) process
- 1.2 Describe how modeling supports the OOSD process
- 1.3 Explain the purpose, activities, and artifacts of the following OOSD workflows: Requirements Gathering, Requirements Analysis, Architecture, Design, Implementation, Testing, and Deployment

Common Course Number: COP 2361C

Unit 2 Object-Oriented Methodologies

General Outcome:

2.0 The student shall:

Understand the basic characteristics of object-oriented systems. Be familiar with the Unified Modeling Language (UML). Comprehend different object-oriented system development (OOSD) methodologies and how to choose the one that is best suited for a particular system.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 2.1 Describe how Object-Oriented principles affect the software development process
- 2.2 Describe the fundamental Object-Oriented principles
- 2.3 Explain the best practices for OOSD methodologies
- 2.4 Describe the features of several common methodologies
- 2.5 Choose a methodology that best suits a particular project

Common Course Number: COP 2361C

Unit 3 Use Case Diagrams

General Outcome:

3.0 The student shall:

Understand how interactions within a system can be modeled using Object Oriented Functional Models. Understand the rules and style guidelines for Activity diagrams, Use Cases, and Use Case Diagrams. Be able to create, document, and validate Functional Models.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 3.1 Identify and describe the essential elements in a UML Use Case diagram
- 3.2 Develop a Use Case diagram for a sample software system
- 3.3 Record Use Case scenarios for architecturally significant Use Cases
- 3.4 Document a Use Case and its scenarios in a Use Case form
- 3.5 Recognize and document Use Case and Actor inheritance
- 3.6 Recognize and document Use Case dependencies
- 3.7 Identify the essential elements in an Activity diagram
- 3.8 Validate a Use Case with an Activity diagram

Common Course Number: COP 2361C

Unit 4 Problem Domain Model - Class Diagrams

General Outcome:

4.0 The student shall:

Understand how the logical organization of data within a system can be modeled using Object-Oriented Structured Models. Understand how CRC cards, Class diagrams, and Object diagrams can be used to model the static structure of a system. Understand the rules and style guidelines for creating CRC cards, Class diagrams, and Object diagrams. Be able to create CRC cards, Class diagrams, and Object Diagrams. Understand the relationship between the Structural and Use Case Models.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 4.1 Identify the essential elements in a UML Class diagram
- 4.2 Construct a Domain model using a Class diagram
- 4.3 Identify the essential elements in a UML Object diagram
- 4.4 Validate the Domain model with one or more Object diagrams
- 4.5 Identify a set of candidate key abstractions
- 4.6 Identify key abstractions using CRC analysis

Common Course Number: COP 2361C

Unit 5 Design Model – Collaboration and Sequence Diagrams

General Outcome:

5.0 The student shall:

Understand how the internal dynamic aspects of a system can be modeled using Object Oriented Behavioral Models. Understand how to use Interaction diagrams such as Sequence and Collaboration diagrams to elaborate the themes of Use Cases in order to address the concept of what an application should do. Understand the relationship between the Behavioral Models and the Structured and Functional Models.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 5.1 Explain the purpose and elements of the Behavioral model
- 5.2 Identify the essential elements of a UML Collaboration diagram
- 5.3 Create a Design model for a Use Case using Robustness analysis
- 5.4 Identify the essential elements of a UML Sequence diagram
- 5.5 Generate a Sequence diagram view of the Design model

Common Course Number: COP 2361C

Unit 6 Fundamental Architecture Concepts

General Outcome:

6.0 The student shall:

Understand how to define the high-level strategy – system architecture, needed for solving a problem and building a solution. Comprehend several common architectural styles, and make a decision about which one is best suited for a particular system.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 6.1 Justify the need for the architect role
- 6.2 Distinguish between architecture and design
- 6.3 Describe the Architecture workflow
- 6.4 Describe the diagrams of the key architecture views
- 6.5 Select the Architecture type
- 6.6 Create the Architecture workflow artifacts

Common Course Number: COP 2361C

Unit 7 Domain Model Refining

General Outcome:

7.0 The student shall:

Understand how to refine the Domain models developed in the Analysis phase in order to produce the needed details for the system design.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 7.1 Refine the attributes of the Domain model
- 7.2 Refine the relationships of the Domain model
- 7.3 Refine the methods of the Domain model
- 7.4 Declare the constructors of the Domain model

Common Course Number: COP 2361C

Unit 8 Design Patterns

General Outcome:

8.0 The student shall:

Understand how available patterns can be reused to solve common software development problems. Comprehend and apply some of the most commonly used patterns.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 8.1 Define the essential elements of a software pattern
- 8.2 Describe the Composite pattern
- 8.3 Describe the Strategy pattern
- 8.4 Describe the Observer pattern
- 8.5 Describe the Abstract Factory pattern

Common Course Number: COP 2361C

Unit 9 State Diagrams

General Outcome:

9.0 The student shall:

Understand how the behavior of dynamic objects represented in a Class diagram can be modeled through State diagrams. Be able to discern the different events within a particular system that will induce an object to change its state.

Specific Measurable Learning Outcomes:

Upon successful completion of this unit, the student shall be able to:

- 9.1 Model object states
- 9.2 Describe techniques for programming complex object states