Course: Object Oriented Design

Semester: Fall

Course Format
And Credit hours: 3 hr Lecture
3 hr Credit

Prerequisites: Experience with a high-level programming language

Instructor: Jeffrey T. Edgell,
304.579.7726,
jeedgell@mix.wvu.edu
Alternate: jeffedgell@comcast.net

Schedule: Monday 6:00

Location: On Line

Office Hours: By appointment

Course Description:

The course will address the fundamental concepts and constructs surrounding object oriented analysis, design, and development. The emphasis however shall be on analysis and design. Comparisons to traditional software development techniques will be made, as appropriate. C++ will be the primary language that is discussed in this course; however the student is free to utilize other object oriented languages. The use of C++ is based on its history and industry popularity. During this course the student will learn to analyze basic problems, formulate solutions, and construct object oriented designs using multiple design techniques.

The course is an essential course in the software engineering curriculum and provides foundational design skills required of all software engineers. The class teaches the student to conduct complex analysis and produce designs in UML. UML is an essential component of the class. UML is a universally recognized object design language and is utilized extensively throughout industry. To enable the student to understand the mapping of design concepts to implementation, C++ is taught at the end of the class. At the completion of the class, a successful student will have the necessary tools to conduct and object oriented analysis, produce a design in UML, and be able to implement that design in C++.

Course Objectives: The objectives of this course are to:
- Evaluate real-world software problem statements and investigate and defend the applicability of an object oriented design approach to minimize and manage complexity
- Author at least three Use Cases for a real world software problem statement.
- Construct a high-level object-oriented design derived from object-oriented analysis resulting
Formulate a detailed design using at a minimum, UML Class, State, Activity, and Interaction diagrams.

Present and defend design decisions amongst evaluating peers as well as critique design solutions proposed by software engineering peers.

Evaluate UML designs and formulate code that realizes the design.

Course Outcomes and Proficiencies:
Upon successful completion of this course:

1. Students will have the ability to identify and appraise the level of complexity related to software engineering problems and formulate design solutions to minimize the complexity.
2. Students will have the ability to weigh the benefits of multiple Object-Oriented Analysis approaches and select and appropriate approach to most efficiently produce the beginning artifacts required to develop an Object-Oriented Design
3. Students will be able to analyze, weigh, and select the best-suited UML diagrams in the development of a design for a given problem.
4. Students will be able to present, argue, and defend solutions they have developed relating to Object-Oriented Analysis and Object-Oriented Design utilizing a variety of approaches with specific emphasis on UML.

Assignments:

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
<th>Point Allocation</th>
<th>Evaluation Criteria</th>
</tr>
</thead>
</table>
| 1          | The purpose of this assignment is to construct a design using more traditional techniques of functional decomposition or action-based decomposition. You may use techniques that you are comfortable with to form your design which may include flow charts, hierarchical design, data flow diagrams, state transition diagrams, pseudo code or any other non object oriented technique you may be familiar with. You may use more than one technique and/or diagram to capture your design. In addition to constructing a design you need to explain why you selected that particular method for the design. Do:  
  o Design definition of the ATM using one or more techniques of your choosing (non object oriented)  
  o An explanation of why you selected each method  
  o You will present your design to the | 100 | You will be evaluated for:  
  o Completeness of design  
  o Correctness of design diagrams used  
  o Presentation and defense of the proposed design  
Grading  
90-100 Correct design, accurate notation, detailed presentation  
80-89 Minor mistakes in design, accurate notation, detailed presentation  
70-79 Minor mistakes in design, minor mistakes in notation, detailed presentation  
60-69 Major mistakes in design, Major mistakes in notation, poor presentation |
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The purpose of this assignment is to produce a high-level class design using the CRC design technique. The basis of the design will be derived from the functional description used in assignment 1 for the ATM. You should identify the classes, the associated responsibilities and collaborators required to achieve this design to complete the design from an Object Oriented perspective.</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Submission Format: PowerPoint</td>
<td>You will be evaluated for:</td>
</tr>
<tr>
<td></td>
<td>Do:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Produce a CRC deck that identifies all the classes in your proposed design in conjunction with the class responsibilities and identified class collaborators to achieve those responsibilities.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o You will present your proposed design in class and also discuss your experience using the CRC process.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>This assignment marks the beginning of an ongoing project that will be used to formulate a complete design for a problem statement given in the class.</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>For this assignment you will construct 3 unique use cases from the problem statement provided and produce 3 independent use case diagrams. The diagrams will be submitted in PowerPoint format. Each use case needs to be complex enough that it can not be explained through basic discussion (i.e. a use case is required for the customer to understand). Each use case will be used in future classes to dictate the context of other UML assignments such as activity diagrams and sequence diagrams in the coming weeks.</td>
<td>You will be evaluated for:</td>
</tr>
<tr>
<td></td>
<td>Submission Format: PowerPoint</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Do:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Submission of 3 use cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Completeness of use cases as related to the problem statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Correctness of diagramming technique for each use case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Presentation and defense of each use case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Grading</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90-100 Correct design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80-89 Minor mistakes in design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70-79 Minor mistakes in design, minor mistakes in notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60-69 Major mistakes in design, Major mistakes in notation, poor presentation</td>
</tr>
</tbody>
</table>
| 4 | For this assignment you will develop and diagram a Class Diagram for the problem scenario given in class. You may utilize whatever method to conduct the object-oriented analysis and conceptualize your design, including the CRC method. You must transform that conceptualization into a UML Class Diagram. You will produce a single diagram. Submission Format: PowerPoint

Do:
- Perform object-oriented analysis on the train management problem statement.
- Produce a UML-based class diagram that represents the class-level design you are proposing.
- Present the class diagram to the class |

| 5 | For this assignment you will develop and diagram UML Activity Diagrams that are derived from the use cases from Assignment 3. You will produce three activity diagrams for this assignment. One for each use case you constructed in the use case assignment. Submission Format: PowerPoint

Do:
- Perform object-oriented/functional analysis on the three use cases for the train management problem statement.
- Produce a three UML-based Activity diagrams, one for each use case. |

| 100 | You will be evaluated for:
- Completeness of the UML class diagram related to the problem statement and use cases
- Correctness of diagramming technique for a UML class diagram
- Presentation and defense of the class diagram

Grading
- **90-100** Correct design, accurate notation, detailed presentation
- **80-89** Minor mistakes in design, accurate notation, detailed presentation
- **70-79** Minor mistakes in design, minor mistakes in notation, detailed presentation
- **60-69** Major mistakes in design, major mistakes in notation, poor presentation

| 100 | You will be evaluated for:
- Submission of 3 Activity Diagrams based on the previously submitted use cases
- Completeness of each UML activity diagram related to the problem statement and use cases
- Correctness of diagramming technique for each UML activity diagram
- Presentation and defense of each activity diagram

Grading
- **90-100** Correct design, accurate notation, detailed presentation

| 4 | 3 | 5 |
| 6 | Select three classes from your current design (or scenarios/use cases) and develop a state diagram for each.  
   Submission Format: PowerPoint  
   Do:  
      o Select three classes or use cases/scenarios from the class scenario  
      o Construct a state diagram for each. | 100 | You will be evaluated for:  
      o Submission of 3 State Diagrams based on three unique classes designed in Assignment 4.  
      o Completeness of each UML state diagram related to the class diagram developed in Assignment 4.  
      o Correctness of diagramming technique for each UML state diagram  
      o Presentation and defense of each state diagram  
   Grading  
   90-100 Correct design, accurate notation, detailed presentation  
   80-89 Minor mistakes in design, accurate notation, detailed presentation  
   70-79 Minor mistakes in design, minor mistakes in notation, detailed presentation  
   60-69 Major mistakes in design, Major mistakes in notation, poor presentation |
| 7 | For this assignment you are to develop and diagram UML Sequence Diagrams and UML Collaboration diagrams based on the Use Cases you derived from Assignment 3 previously given. You will produce three Interaction diagrams and three Sequence diagrams for this assignment. One Sequence and one Collaboration (based on the same Use Case) for each use case you constructed in the use case assignment.  
   Submission Format: PowerPoint  
   Do:  
      o Produce a three UML-based Sequence | 100 | You will be evaluated for:  
      o Submission of 3 Sequence Diagrams based on three unique Use cases.  
      o Submission of 3 interaction Diagrams based on three unique Use cases.  
      o Completeness of each UML sequence diagram related to the each use case.  
      o Completeness of each UML interaction
<table>
<thead>
<tr>
<th></th>
<th>Produce a three UML-based Collaboration diagrams, one for each use case.</th>
<th>Correctness of each UML sequence diagram related to the each use case.</th>
<th>Grading</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td></td>
<td>O</td>
<td><strong>90-100</strong> Correct design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td>O</td>
<td></td>
<td>O</td>
<td><strong>80-89</strong> Minor mistakes in design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td>O</td>
<td></td>
<td>O</td>
<td><strong>70-79</strong> Minor mistakes in design, minor mistakes in notation, detailed presentation</td>
</tr>
<tr>
<td>O</td>
<td></td>
<td>O</td>
<td><strong>60-69</strong> Major mistakes in design, major mistakes in notation, poor presentation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th>Provide a consolidated design package of all products developed in support of the entire design scenario given in Assignment 3. This includes: 1) Use Cases 2) Class Diagram 3) Activity Diagrams 4) State Diagram 5) Interaction Diagrams</th>
<th>You will be evaluated for:</th>
<th><strong>Grading</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Submission Format: PowerPoint</td>
<td></td>
<td><strong>90-100</strong> Correct design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td>Do:  o Review and document all intended changes for the final consolidated project. o Provide a summary of all changes made to the original design documents along with an explanation as to why the changes were made.</td>
<td></td>
<td><strong>80-89</strong> Minor mistakes in design, accurate notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>70-79</strong> Minor mistakes in design, minor mistakes in notation, detailed presentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>60-69</strong> Major mistakes in design, major mistakes in notation, poor presentation</td>
</tr>
</tbody>
</table>

Note that all projects are equally weighted.
Text:

Object-Oriented Design and Analysis with Applications, Third Addition

Fowler

Grading:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>35 %</td>
</tr>
<tr>
<td>Final</td>
<td>40 %</td>
</tr>
<tr>
<td>Project and Assignments</td>
<td>25 %</td>
</tr>
<tr>
<td></td>
<td>100 %</td>
</tr>
</tbody>
</table>

Grade Assignment:

100 – 90 A
89 – 80 B
79 – 70 C
69 – 60 D
59 – 0 F

Tests:

All tests are designed to gauge the student’s understanding of topics covered in assigned reading, lectures, and homework assignments. A midterm and final will be given. The final will be comprehensive. All tests will be graded, returned, and discussed one week following submission. A general review of each question will be conducted in the class following the submission.

Grading Policy:

Make-up exams may be given under special circumstances. It is your responsibility to arrange for a make-up prior to the examination.

Assignments may be handed in late under special circumstances. It is the student’s responsibility to negotiate the terms of an extension to an assignment. Course policy is a reduction of 20% per class that the assignment is late.

HW Assignments:

Homework assignments will be given throughout the semester and are worth approximately the same credit. The assignments will be used to emphasize lecture material and produce artifacts for a group project. The majority of assigned projects take place in a group setting to emphasize real world complexities in software engineering. All homework assignments will be graded, returned, and discussed one week following submission. A general review of each question will be conducted in the class following the submission.
**Design Project:** During this course, you are expected to work in a group of 2 or 5 students to analyze a given problem, utilize analysis techniques discussed in class, document the proposed design utilizing UML, and present the proposed design. The problem statement will be given to you in approximately the 3rd class of the semester. The project will be assembled throughout the duration of the class resulting in a Use Case, Class Diagrams, Activity Diagrams, State Transition Diagram, Sequence Diagrams, Collaboration Diagrams, and Package Diagrams. You and your group members will be required to present the design solution in a formal oral presentation, in front of faculty and students, at an assigned time in the semester. Because of time constraints, each group will only present portions of the design, as assigned by the instructor. All segments of the design project will be graded, returned, and discussed one week following submission. A general review of each question will be conducted in the class following the submission.

**Attendance Policy:** Consistent with WVU guidelines, students absent from regularly scheduled examinations because of authorized University activities will have the opportunity to take them at an alternate time. Make-up exams for absences due to any other reason will be at the discretion of the instructor.

**Other Policy and Support:** Please refer to [https://ecampus.wvu.edu/webapps/blackboard/content/listContentEditable.jsp?content_id=_2489792_1&course_id=_48260_1](https://ecampus.wvu.edu/webapps/blackboard/content/listContentEditable.jsp?content_id=_2489792_1&course_id=_48260_1) for all other WVU policies and support that is important for you to be aware of for this course as well as others at WVU.
<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
<th>Readings/Assignment</th>
<th>Assignment Due/Returned</th>
<th>Module Objective (MO)</th>
<th>Class Objective (CO)</th>
</tr>
</thead>
</table>
| 1    | Complexity in Design  
|       | - Real world complexity  
|       | - Sources of complexity  
|       | - An approach to minimizing complexity | Read  
|       | - Module 1  
|       | - Module 1 class Slides  
|       | - Object-Oriented Design and Analysis with Applications: Chapter 1 | 1.
|       | Do  
|       | - Participate in threaded discussion on complexity (CO 1; MO 1, 2, 3)  
|       | - Participate/review online lecture (CO 1; MO 1, 2, 3) | 1. Evaluate the aspects of complexity in contemporary software  
|       | 2. Examine the key contributors of complexity to software design  
|       | 3. Critique design approaches and how they may or may not assist in minimizing complexity. | 1. Evaluate real-world software problem statements and investigate and defend the applicability of an object oriented design approach to minimize and manage complexity |
| 2    | The Structure of Design  
|       | - Action-based design/functional decomposition  
|       | - Object-oriented analysis and design  
|       | - Design validation  
|       | - Review ATM Functional Design Requirement | Read  
|       | - Module 2  
|       | - Module 2 class slides  
|       | - Participate in threaded discussion | Due: Week 3  
|       | Do  
|       | - Participate/review online lecture (CO 1; MO 1, 2, 3)  
|       | Assignment 1: Form teams and produce a functional design for an Automated Teller Machine and present in class. (CO 1; MO 1, 2) | Returned and Reviewed: Week 4 |
| 3    | The Object Concept  
|       | - The evolution of design and languages  
|       | - Object Oriented-Design paradigm  
|       | - Data design  
|       | - What are objects and classes  
|       | - Class presentations for the ATM functional design | Read  
|       | - Module 3  
|       | - Module 3 class slides  
|       | - Participate in threaded discussion | 1. Differentiate among the object design approach as it compares to other design approaches.  
|       | Do  
|       | - Participate/review online lecture (CO 1, 2; MO 1, 2, 3)  
|       | - Present Assignment 1 (CO 1, 2; MO 2, 3)  
|       | Alternate Resource  
|       | - Alternate resource: Using CRC Cards: An Informal Approach to Object-Oriented Design ISBN 0133746798 | 2. Evaluate and critique class presentations of Assignment 1  
|       | 3. Develop an object-oriented design using Assignment 1 as the basis (ATM) using the CRC technique. | 1. Construct a high-level object-oriented design derived from object-oriented analysis resulting in a class diagram with defined relationships and cardinality.  
<p>|       | 2. Present and defend design decisions amongst evaluating peers as well as critique design solutions proposed by software engineering peers. | 1. Formulate a detailed design using at a minimum, UML Class, |</p>
<table>
<thead>
<tr>
<th>Course: UML Foundations</th>
<th>Assignment Details</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module 5:</strong> Basic Object-Oriented Design Approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Objective:</strong> Analyze UML diagrams and their components</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Module 6:</strong> Types of Classes and Relationships</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Objective:</strong> Classify different types of classes and relationships</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Module 7:</strong> UML Foundations</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Objective:</strong> Understand UML notation and its usage</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Module 8:</strong> Class Diagrams</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Objective:</strong> Create and interpret class diagrams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements for Class Diagrams and Activity Diagrams and Collaboration Diagrams</td>
<td>Participation in online lecture (MO 2,3,4)</td>
<td>Participation in threaded discussion (MO 2,3,4)</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Present a UML Class Diagram based on provided requirements. (CO 1,2; MO 1,3,4)</td>
<td>Assignment 4: Produce and present a UML Class Diagram based on provided requirements. (CO 1,2; MO 1,3,4)</td>
<td>Present and defend state diagrams</td>
</tr>
<tr>
<td>Midterm Due: Week 10</td>
<td>Midterm returned and reviewed: Week 11</td>
<td>Midterm returned and reviewed: Week 11</td>
</tr>
<tr>
<td>Assignment 5 Due: Week 11</td>
<td>Assignment 5 Returned and Reviewed: Week 12</td>
<td>Assignment 5 Returned and Reviewed: Week 12</td>
</tr>
</tbody>
</table>

9 Class Diagram Extensions and Activity Diagrams
- Modules:
  - Module 9
  - Module 9 class slides
  - UML Distilled: Chapter 11
  - IBM’s Overview of Class and Extended Class Diagrams
  - Rational Edge Activity Diagrams

Assignments:
- Assignment 5: Produce and present three Activity Diagrams based on Use Cases developed from Assignment 3 from the provided requirements (CO 1,2; MO 1,3,4)
- Assignment 6: Produce and present a UML Class Diagram based on provided requirements (CO 1,2; MO 1,3,4)

Midterm Due: Week 10
Midterm returned and reviewed: Week 11
Assignment 5 Due: Week 11
Assignment 5 Returned and Reviewed: Week 12

10 State Transition Diagrams
- Modules:
  - Module 10
  - Module 10 class slides
  - UML Distilled: Chapter 10
  - Read State Machine Theory
  - UML State Transition Diagrams Overview
  - Read Class Slides

Assignments:
- Assignment 5: Produce and present three UML State Diagrams based on provided requirements of three classes selected from Assignment 4. (CO 1,2; MO 1,3,4)
- Assignment 6: Produce and present a UML State Diagram based on provided requirements (CO 1,2; MO 1,3,4)

Due: Week 12
Returned and reviewed: Week 13

11 Interaction Diagrams
- Modules:
  - Module 11
  - Module 11 class slides

Assignments:
- Assignment 7: Produce and present a UML State Diagram based on provided requirements (CO 1,2; MO 1,3,4)
- Assignment 8: Produce a collaboration diagram and a sequence diagram based on a collaboration diagram and a sequence diagram

Due: Week 13
Returned and reviewed: Week 14

1. Present and defend state diagrams
2. Argue the need for collaboration and sequence diagrams in a design
3. Compare and contrast the information provided in an collaboration diagram and a sequence diagram
4. Argue what audience is appropriate for state diagrams
5. Investigate a problem statement and formulate a detailed UML state diagram

1. Present and defend activity diagrams
2. Compare and contrast state diagrams to state transition and finite state machine diagramming techniques
3. Relate UML state diagrams to procedural and function-oriented design
4. Argue what audience is appropriate for state diagrams
5. Investigate a problem statement and formulate a detailed UML state diagram

1. Present and defend design decisions amongst evaluating peers as well as critique design solutions proposed by software engineering peers.
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
<th>Read</th>
<th>Do</th>
<th>Postgraduate</th>
<th>Undergraduate</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Wrapping Up UML with Remaining Diagrams</td>
<td>Read o Module 12 o Module 12 class slides Do o Participate in online lecture (MO 1,2,3,4) o Participate in threaded discussion (MO 1,2,3) o Present sequence and collaboration diagrams (CO 1,2; MO 1,4)</td>
<td>Present and defend sequence and collaboration diagrams 2. Evaluate and select for what purpose component, deployment, object, composite structure, interaction overview, timing, state and value lifeline diagrams should be used. 3. Argue what audience is appropriate for each diagram 4. Compare and contrast the information provided in an a collaboration diagram and a sequence diagram</td>
<td>1. Present and defend sequence and collaboration diagrams 2. Evaluate and select for what purpose component, deployment, object, composite structure, interaction overview, timing, state and value lifeline diagrams should be used. 3. Argue what audience is appropriate for each diagram 4. Compare and contrast the information provided in an a collaboration diagram and a sequence diagram</td>
<td>1. Formulate a detailed design using at a minimum, UML Class, State, Activity, and Interaction diagrams 2. Present and defend design decisions amongst evaluating peers as well as critique design solutions proposed by software engineering peers.</td>
</tr>
<tr>
<td>13</td>
<td>Realizing Design to Implementation</td>
<td>Read o Module 13 o Module 13 class slides o Read Bjoern Stroustrup’s C++ Primer Do o Participate in online lecture (CO 1; MO 1,2,3) o Participate in threaded discussion (CO 1; MO 1,2)</td>
<td>Investigate what UML diagrams can be used in the formulation of executable code for functional and procedural content 2. Translate algorithmic design and to appropriately formed logical expressions 3. Design code that translates design actions documented in activity and sequence diagrams</td>
<td>1. Investigate what UML diagrams can be used in the formulation of executable code for functional and procedural content 2. Translate algorithmic design and to appropriately formed logical expressions 3. Design code that translates design actions documented in activity and sequence diagrams</td>
<td>1. Evaluate UML designs and formulate code that realizes the design</td>
</tr>
<tr>
<td>14</td>
<td>Class Implementation from Design</td>
<td>Read o Module 14 o Module 14 class slides Do o Participate in online lecture (CO 1; MO 1) o Participate in threaded discussion (CO 1; MO 1)</td>
<td>Develop code from class diagrams and state diagrams to produce code that results in a well defined class interface</td>
<td>1. Develop code from class diagrams and state diagrams to produce code that results in a well defined class interface</td>
<td>1. Evaluate UML designs and formulate code that realizes the design</td>
</tr>
<tr>
<td>15</td>
<td>Class Implementation from Design Continued</td>
<td>Read o Module 15 o Module 15 class slides Do o Participate in online lecture (CO 1; MO 1,2,3) o Participate in threaded discussion (CO 1; MO 1,2,3) Assignment Assignment: Complete UML design package for submission. The package is Due: Week 16 Returned and Reviewed: Week 17</td>
<td>Develop code from all UML diagrams that produces a complete class 2. Develop code from all UML diagrams that defines all functions of a class 3. Develop code from all UML diagrams that implements all the functions of a class</td>
<td>1. Develop code from all UML diagrams that produces a complete class 2. Develop code from all UML diagrams that defines all functions of a class 3. Develop code from all UML diagrams that implements all the functions of a class</td>
<td>1. Evaluate UML designs and formulate code that realizes the design 2. Formulate a detailed design using at a minimum, UML Class, State, Activity, and Interaction diagrams</td>
</tr>
</tbody>
</table>
the entire set of assignments from the Use Case Assignment (#3) to the Interaction Diagrams Assignment (#7) listing any changes and adjustments made through the course of the class as you better understood your design. (CO 2)

| 16 | Final Overview  
|    | o Review and summary of the semester  
|    | o Final exam and project review  
|    | o Questions and Answers  
| Read | o Module 16  
|      | o Module 16 class slides  
| Do  | o Participate in online lecture . (CO 1; MO 1,2,3,4,5)  
|     | o Participate in threaded discussion . (CO 1; MO 1,2,3,4,5)  
| Assignment | o Final  
| Due: Week 17 | 1. Weigh the elements of complexity  
|          | 2. Argue and defend the object model  
|          | 3. Argue the object model and how it minimizes complexity  
|          | 4. Compare and contrast UML to other design techniques  
|          | 5. Differentiate among object and non object-oriented languages  
|        | 1. All |