

CS3331: Advanced Object-Oriented Programming

January 14, 2015

Level 1: Knowledge and Comprehension (Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions.) The material has been presented only at a superficial level. Upon successful completion of this course, students will be able to:

- 1a. Explain the differences between an object-oriented approach and a procedural approach.

Level 2: Application and Analysis (Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.) Upon successful completion of this course, students will be able to:

- 2a. Formulate use-case diagrams and scenarios to support understanding of user requirements.
- 2b. Use object-oriented design notations, including UML class diagrams, state machine diagrams, and sequence diagrams, to model problem solutions.
- 2c. Use basic object-oriented design patterns to structure solutions to software design problems.
- 2d. Translate design features, such as associations, relationships and multiplicities, to implementations.
- 2e. Use frameworks and library classes and methods in problem solutions.

Level 3: Synthesis and Evaluation (Level 3 outcomes are those in which the students can apply the material in new situations. This is the highest level of mastery.) Upon successful completion of this course, students will be able to:

- 3a. Design and implement software employing the principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.
- 3b. Design, implement, and use classes and methods that follow conventions and styles, and make appropriate use of advanced features such as inheritance, exception handling, and generics.
- 3c. Evaluate existing classes and software for the purposes of extension through inheritance.
- 3d. Create API documents for classes, fields and methods.
- 3e. Design and implement test suites for automated unit testing.
- 3f. Re-factor existing software to improve its design or efficiency.