Course number and name: CS 04113: Introduction to Object Oriented Programming

Credits and contact hours: 4 credits / 5 contact hours

Instructor’s or course coordinator’s name: Chia Chien


Specific course information

Catalog description: This course introduces the fundamental concepts of programming from an object-oriented perspective. Topics are drawn from classes and objects, abstraction, encapsulation, data types, calling methods and passing parameters decisions, loops, arrays and collections, documentation, testing and debugging, exceptions, design issues, inheritance and polymorphic variables and methods. The course emphasizes modern software engineering and design. Students are expected to be sufficiently proficient in mathematics such that they are ready to take Calculus I (MATH 01.130).

Prerequisites: MATH 01122 Precalculus Mathematics or MATH 01130 Calculus I

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Specific goals for the course

1. **object-oriented principles.** Students have demonstrated mastery of object-orientation through examination assessments on such concepts.
   - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
   - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

2. **introductory Java programming.** Students have effectively applied Java programming structures to implement coding assignments and to complete test assessments.
   - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
   - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
   - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice
o ABET (k) An ability to apply design and development principles in the construction of software systems of varying complexity

3. **fundamental project development.** Students have successfully utilized an integrated development environment to design, implement, document, test, and execute applications.

   o ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
   o ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
   o ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice
   o ABET (k) An ability to apply design and development principles in the construction of software systems of varying complexity

Required list of topics to be covered

1. Class structure and interactions
2. Primitive and object data types
3. Conditional structures (i.e. if-else, switch)
4. Iterative loops (i.e. for-each, for, while, do-while)
5. Operators and expressions
6. Class and object diagrams
7. Documentation and style
8. Testing and debugging
9. Integrated developing environment
10. Object-oriented principles (i.e. abstraction, encapsulation, coupling, cohesion)
11. Inheritance and polymorphic variables
12. Arrays, ArrayLists, HashMaps, HashSets

Optional list of topics that could be covered

13. Streams
14. Functional Processing
15. Multi-dimensional arrays