LEARNING COMMONS

FRANKLIN UNIVERSITY PROFICIENCY EXAM (FUPE)
STUDY GUIDE

Course Title: COMP 111: Introduction to Computer Science & Object-Oriented Programming

Recommended Textbook(s): https://www.franklin.edu/current-students/academic-resources/textbooks

Number & Type of Questions: N/A

Permitted Materials: No Materials Permitted

Time Limit: 120 minutes (2 hours)

Minimum Passing Score: 75%

Format varies

Outline of the Topics Covered:

Course Description

This course provides an introduction to software construction using an object-oriented approach. The student learns and reflects on problem analysis, object-oriented design, implementation, and testing. To support the concepts and principles of software construction, the student will design, code, test, debug, and document programs using the Java programming language. Basic data types, control structures, methods, and classes are used as the building blocks for reusable software components. Automated unit testing, programming style, and industrial practice are emphasized in addition to the object-oriented techniques of abstraction, encapsulation, and composition.

Prerequisites

MATH 160: College Algebra

Course Outcomes

Upon successful completion of this course, students will be able to:

1. Apply the object-oriented principles of encapsulation, composition, and abstraction to analyze problems and design solutions.
2. Use classes, objects, and methods to implement object-oriented designs in Java.
3. Identify and use appropriate data types and control structures within class and method implementations.
4. Incrementally test and document implementations using industry accepted approaches.
5. Discuss relationships between the disparate topic areas addressed in this course.

Course Content

- Week 1: Edit, compile, run. Variables and types. Syntax and logic errors.
- Week 2: Classes, objects, and methods. Accessor and mutators. Unit testing introduction.
- Week 3: Encapsulation and abstraction. Implement classes and methods. Document APIs.
- Week 4: Variable scope and lifetime. Method parameters. Primitive vs. class types.

Strings. Simple I/O.

- Week 5: Selection control structures. Relational operators. Logical operators. Boolean expressions.
- Week 6: Test driven development. Test case coverage. Black-box, white-box, and regression testing.
- Week 7: Repetition control structures. Loop errors (off by one, infinite). Sentinels, counting, and symmetric vs. asymmetric bounds.
- Week 8: Random numbers. Debugging. Arrays and Array List data structures. Array algorithms.
- Week 9: Multi-dimensional arrays. Wrapper classes. Quadratic sorting algorithms. Linear vs. binary search.
- Week 10: Object oriented analysis and design. Coupling and cohesion. Advanced array algorithms.
- Week 11: Pre- and post-conditions. Static keyword. Introduction to UML diagrams.
- Week 12: Review and final exam.