LEARNING COMMONS

FRANKLIN UNIVERSITY PROFICIENCY EXAM (FUPE)
STUDY GUIDE

Course Title: COMP 121 Introduction to Computer Science and Object-Oriented Programming

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

Number & Type of Questions: 50 - True/false, multiple choice, short answer, definitions, and code writing.

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 continues the objected-oriented approach to software construction. The student learns and reflects on advanced object-oriented techniques, algorithm efficiency, class hierarchies, and data structures. To support the concepts and principles of software construction, the student will design, code, test, debug, and document programs using the Java programming language. Design principles, I/O, exception handling, linear data structures (lists, stacks, and queues), and design patterns are emphasized in addition to the object-oriented techniques of inheritance and polymorphism.

Prerequisites

COMP 111: Introduction to Computer Science and Object-Oriented Programming

Course Outcomes

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

1. Apply the advanced object-oriented principles of inheritance and polymorphism to analyze real-world problems and design, implement, and test solutions.
2. Detect and correct runtime program errors using exceptions.
3. Read and write random access and sequential files of text and binary data.
4. Analyze space and time complexity of algorithms.
5. Design, implement, test, debug, and document linear data structures.
6. Use lists, stacks, and queues to solve problems.
7. Recognize and apply the design patterns of Strategy, Template Method, Iterator, and Adapter in object-oriented designs.
8. Discuss relationships between the disparate topic areas addressed in this course.

Course Content

- Week 1: Objects/Classes
- Week 2: Abstract Classes and Interfaces
- Week 3: Strategy and Template
- Week 4: Files and Exception
- Week 5: Software Development Life Cycle and Efficiency
- Week 6: Lists and Collections, Iterator
- Week 7: Abstract List, Generic Programming
- Week 8: Linked Collection/Adapter

- Week 9: Linked List/Decorator
- Week 10: Stacks
- Week 11: Queues
- Week 12: Final Exam and Reflection