LEARNING COMMONS

FRANKLIN UNIVERSITY PROFICIENCY EXAM (FUPE)
STUDY GUIDE

Course Title: COMP 101: Problem Solving with Computing

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

Number & Type of Questions: There are 4 pages to this exam. Each page is a different section. The first page is worth 25 points and includes true/false, multiple choice, and fill-ins (25 questions at 1 point each). The second page is short answer questions. It includes 10 questions at 3 points each. The third page is problems and includes 25 questions at 1 point each. The fourth page is programming. It includes 4 programming problems at 10 points each.

Permitted Materials: No Materials Permitted

Time Limit: 120 minutes (2 hours)

Minimum Passing Score: 75%

Language: The exam currently uses Python 3 as its programming language. A review of this language would be appropriate and is strongly suggested.

Notes: This is a difficult exam. You will be expected to do problems related to both reading and writing code. The outcomes below contain more detail about the exam contents.

Format varies

Outline of the Topics Covered:

Course Description

Many organizations today utilize computers and information systems to store, organize, analyze, and summarize data to solve problems. As a result, computing is a tool that can benefit students in many different fields. At the heart of solving problems with computers is the study of structured thinking using algorithms. This course is designed for students with no prior programming experience and teaches the building blocks of algorithms, including variables, expressions, selection and repetition structures, functions and parameters, and array processing.

Course Outcomes

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

1. Describe what a program is, how to write a program, and how to handle program errors.
2. Produce a working program that solves a significant problem.
3. Test and debug programs.
4. Employ modularization techniques in files and functions.
5. Employ existing libraries to create larger programs.
6. Use relational and Boolean operators to control the flow of a program.
7. Recognize the use of and apply iteration for repeated control flow.
8. Learn how to formulate problems, think creatively about solutions, and express a solution clearly and accurately.
9. Explore the breadth of impact of computer science in broader society.