COMP 101 Study Guide for FUPE

Format

The FUPE exam for COMP101 is an online exam. You will have up to 2 1/2 hours (150 minutes) to take the exam. It is worth up to 125 points.   You must pass this FUPE with an 80% on it in order to pass.

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.

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.

Language

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

Recommended textbook and other resources

How to Think Like a Computer Scientist: Interactive Edition

This free interactive book (3rd edition) is a product of the Runestone Interactive Project at Luther College, led by Brad Miller and David Ranum.

NOTE: You will be required to register and create a log in to use and get access to this textbook. Please be sure to enter COMP101 as the course name when registering.

Use of the Repl.it website is recommended that uses the Python 3 programming language. You will be provided this link during the exam to complete your programming problems at the end so it is suggested you open it and become familiar with it prior to taking the assessment.

A byte of Python - a free online textbook for very beginners.
Non-Programmer's Tutorial for Python 3 a free wikibook available in a printable and PDF versions.

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.