McMaster University
SFWR ENG 2MD3 Winter 2019 Assignment 3
Due: Sunday March 1, 2020 at 23:55

February 24, 2020

Name: Manuel Lemos ID: 400177763

Provide answer to questions 1 to 5 of Review Questions 4.6 of the book. [10(2 marks each)]

1. Give two synonyms for the word module, used when referring to entities similar to modules in other contemporary programming languages (such as Pascal or Ada).

   The terms unit, package and assembly are used when referring to entities similar to modules

2. What is procedural abstraction?

   Procedural abstraction is the concept of separating the implementation of a method from the users. This results in one being able to make use of a methods functionality without understanding its implementation.

   As a result, when one wishes to change the way in which a method is implemented, a change to every occurrence of such a method is not necessary.

3. What is data abstraction?

   Data abstraction is the concept of hiding the details of a data structures implementation and the representation of the data structure itself.

   Once again, the result of abstraction is knowing only what is done and not how the details are implemented.

4. What is encapsulation?

   Encapsulation can be achieved by hiding features or entire program components inside of a ‘capsule’ such that, only once inside the capsule can they be seen and used.

5. Summarize the benefits of using C modules properly.

   - Separated Compilation of program components
   - Program modifications are easier to enact
   - Program is better organized
   - Abstraction, encapsulation, and information-hiding
   - Promotes software reuse

Question 2 (10 marks)

1. Provide answer to question 3 of Review Questions 5.2 of the book. [2 marks]

What is stepwise refinement? What gets refined? What are the steps in stepwise refinement?

> The concept of creating and designing software by beginning at the highest level of abstraction, which may consist of simple design ideas, and moving to lower levels of abstraction until the lowest level of abstraction is reached and the software has been entirely developed.
>
> Effectively, a series of refinements are enacted on low detail, high level abstraction program ideas until a full program is developed.

2. Provide answer to questions 1 to 4 of Review Questions 5.5 of the book. [8(2 marks each)]

1. What is the difference between unit testing and integration testing?

   Unit testing is individually testing functions on their own to ensure they work for all possible inputs.

   Integration testing is testing functions acting together as they would under normal operation within the context of functional program.

2. Explain the concepts of regression testing and acceptance testing.

   Regression testing is used when something about the program has been changed. Once the change has been implemented, a complete retest of the program, to ensure continued functionality, is done.

   Acceptance testing is when a completed piece of software undergoes a set of tests specified by the client. To be accepted by the client, the software must pass the tests.

3. What is the difference between bottom-up testing and top-down testing?

   Bottom up testing consists of testing the lowest level components of a system first and then testing the other parts of the system which use them next.

   Top down testing consists of testing the highest level components of a system first which use other components of a system, followed then, by testing the lower level components these high level components use.

4. What are stubs and test drivers?

   A stub is a 'subroutine' which stands in place for a piece of code which will later be written. This subroutine fakes the correct output for a test input case.

   A test driver is module which acts as a means to call the temporary subroutine (stub), and check that the subroutine outputs the correct value.

Question 3 (11 marks)

1. Provide answer to questions 1 to 4 of Review Questions 5.6 of the book. [8(2 marks each)]

1. Explain the philosophy of measurement and tuning?

   It is difficult to look at a program and determine which components are inefficient. Thus, one must first measure where a system spends the largest amounts of time, and then tune these components to reduce execution time and improve program efficiency.

2. What is an execution-time profile tool (or a profiler, for short)?

   A profiler is a tool which measures where a program spends its time during execution. This information is then translated and displayed as percentages of net time spent on a program element.

3. Does experience show that most programs consume running time approximately uniformly throughout their code, or is the distribution of time consumed often more uneven:

   Most programs consume run time in a way which is uneven throughout the code.

4. What do you do when tuning a program to make it more efficient?

   Tuning consists of rewriting inefficient code components, attempting to improve efficiency, and then testing the rewritten code to see if it is improved or not.

2. Provide answer to question 1 of Exercises 5.6 of the textbook. [3 marks]

Find the inefficiency in the following program, and then tune it to improve its running time.

The line: **Middle = (int) floor((m + n) / 2.0);** Is extremely inefficient.

Floating point division is forced by using 2.0 as a denominator. Then the floating point is rounded down to the nearest integer using the floor function, then the value is casted to int.

Optimized version: **Middle = (m+n)/2;** This way, floating point division and use of the floor function is avoided. As a result, a more efficient solution is achieved.

Question 4 (4 marks)

1. Provide answer to question 3 of Review questions 5.7 of the book. [2 marks]

What is bottom-up programming? When might it be useful to plan to implement a software system using bottom-up programming?

> The concept of bottom up programming is to construct and implement pieces of software which are general purpose and modular. Then, using these components, one can assemble a higher-level system.
>
> Bottom up programming is useful when developing reusable libraries as it enables one to reuse code for future projects thus saving time and money.

2. Provide answer to question 1 of Exercises 5.7 of the textbook. [2 marks]

A software system, called VisiPhysics, to be used for visualizing physics data, is estimated to require 56,000 lines of source code to implement starting from scratch, but it is estimated to require only 32,000 lines of extra source code to be implemented starting with two numerical and graphics subroutine libraries. If programmer time costs $100,000 per PY and the two libraries cost $150,000 each to purchase, is it cheaper to implement VisiPhysics by starting from scratch, or by buying and using the two libraries? (Assume that Equation 5.1 accurately pre- dicts the effort needed on the VisiPhysics project.)

$PM = 2.4*(KDSI)^{1.05}$

Create from Scratch: $\$100{,}000(2.4*(56)^{1.05})/12 = \$1{,}369{,}706.21$

Create with Library: $\$150{,}000 * 2 + \$100{,}000(2.4*(32)^{1.05})/12 = \$1{,}061{,}092.55$

As a result, using the libraries saves over $300,000