**Name: Kenyon Geetings**

# CMSC 115 Reading Guide

Please enter your responses in red.

## Chapter 3

## Section 3.1 Algorithms Skim

1. What do their recipe and their Babylonian square root algorithm have in common? How are they different? A recipe and a Babylonian square root algorithm both have a lack of details in common. They are different in that one gives you a first guess on how much of each ingredient you will need while the other asks for you to provide your own guess.

## Section 3.2 Algorithm Features Read

1. What is the distinction between an algorithm and a program? The distinction between an algorithm and a program is that an algorithm is a description of how a problem can be solved, while a program is an implementation of an algorithm in a particular language on a particular kind of computer.
2. Why do they suggest separating *what to do* from *how to do it* in your mind? The recommend separating what to do from how to do it in your mind as it provides a couple of advantages. It allows you to analyze the algorithm itself, it allows you to see the strengths and weaknesses of a certain programming language, and you can analyze the impact of a particular computer and language on the implementation.
3. What qualities should an ideal algorithm have? An ideal algorithm should be detailed, effective, specific as to its behavior, and have a general purpose.

## Section 3.3 What is a Program? Read

1. What qualities should an ideal program have? An ideal program should be readable, robust, and have correctness.
2. How do these qualities differ from (and are perhaps in conflict with) the qualities for an ideal algorithm? These qualities differ and conflict with the qualities of an ideal algorithm as an algorithm is an abstraction of a solution to a problem, and the program is the actual solution. The qualities of a program are more geared for a human to understand than from a computer.
3. Why is good naming such an important practice? Good naming is such an important practice because it helps with readability and it is relatively easy to do.
4. What is *robustness*? Robustness is what a program should strive to be. It means that a computer will not break down if it encounters some input that it does not understand or anticipate.