***Pseudocode\_Algorithms\_Examples***

*Pseudocode* is a cross between human language and a programming language. Although the computer can’t understand pseudocode, programmers often find it helpful to write an algorithm in a language that’s “almost” a programming language, but still very similar to natural language. For example, here is pseudocode that describes the pay-calculating program:

*Get payroll data.*

*Calculate gross pay.*

*Display gross pay.*

(Gaddis 20)

Gaddis, Tony. *Starting Out with C++, 9th Edition*. Pearson, ISBN 9780134498379, 2017

**VideoNote**

**Designing a Program with Pseudocode**

(Gaddis 20)

Gaddis, Tony. *Starting Out with C++, 9th Edition*. Pearson, ISBN 9780134498379, 2017

Although the pseudocode above gives a broad view of the program, it doesn’t reveal all the program’s details. A more detailed version of the pseudocode follows.

*Display “How many hours did you work?”.*

*Input hours.*

*Display “How much do you get paid per hour?”.*

*Input rate.*

*Store the value of hours times rate in the pay variable.*

*Display the value in the pay variable.*

Notice the pseudocode contains statements that look more like commands than the English statements that describe the algorithm in [Section 1.4](http://e.pub/gro1c2m5ar5y2071jgty.vbk/OPS/xhtml/fileP70004943860000000000000000007F3.xhtml#P70004943860000000000000000007F3) (What Is a Program Made of?). The pseudocode even names variables and describes mathematical operations.

(Gaddis 20-21)

Gaddis, Tony. *Starting Out with C++, 9th Edition*. Pearson, ISBN 9780134498379, 2017

**Focus on Problem Solving: A Case Study**

(Gaddis 134-138)

Gaddis, Tony. *Starting Out with C++, 9th Edition*. Pearson, ISBN 9780134498379, 2017

**Program Design**

The program must perform the following general steps:

1. Ask the user to enter the dimensions of the crate (the crate’s length, width, and height).
2. Calculate the crate’s volume, the cost of building the crate, the customer’s charge, and the profit made.
3. Display the data calculated in Step 2.

A general hierarchy chart for this program is shown in [Figure 3-7](http://e.pub/gro1c2m5ar5y2071jgty.vbk/OPS/xhtml/fileP7000494386000000000000000001ACC.xhtml#P7000494386000000000000000001B08). See also fig 3.8, 3.9, 3.10

Pseudocode for the program is as follows:

*Ask the user to input the crate's length.*

*Ask the user to input the crate's width.*

*Ask the user to input the crate's height.*

*Calculate the crate's volume.*

*Calculate the cost of building the crate.*

*Calculate the customer's charge for the crate.*

*Calculate the profit made from the crate.*

*Display the crate's volume.*

*Display the cost of building the crate.*

*Display the customer's charge for the crate.*

*Display the profit made from the crate.*

**Calculations**

The following formulas will be used to calculate the crate’s volume, cost, charge, and profit:

* volume = length × width × height
* cost = volume × 0.23
* charge = volume × 0.5
* profit = charge − cost

***Creating Algorithm in Programming:***

[http://www.programiz.com/article/algorithm-programming](http://www.programiz.com/article/algorithm-programming" \t "_parent)

"In programming, algorithm are the set of well defined instruction in sequence to solve a program. An algorithm should always have a clear stopping point.

**Qualities of an good algorithm**

1. Inputs and outputs should be defined precisely.
2. Each steps in algorithm should be clear and unambiguous.
3. Algorithm should be most effective among many different ways to solve a problem.
4. An algorithm shouldn't have computer code. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages."