Pseudocode Guidelines

Pseudocode is a kind of structured English for describing algorithms. It allows the designer to focus on the logic of the algorithm without being distracted by details of language syntax.  The pseudocode needs to be complete and describe the entire logic of the algorithm so that implementation becomes a rote mechanical task of translating line by line into source code.

The vocabulary used in the pseudocode should **NOT** be the vocabulary of the implementation domain (i.e Java, C++).  The pseudocode is a narrative for someone who knows the requirements and is trying to learn how the solution is organized. Write the pseudocode as if the person reading it knows nothing about Java or computer science. There should be no mention of classes, objects, variables, scanners, system.in, etc.

Each of the steps of the pseudocode may be numbered for clarity and to help with repetition statements if needed.

The logic must be decomposed to the level of a single loop or decision. Thus "Search the list and find the customer with highest balance" is too vague because it takes a loop AND a nested decision to implement it.

Three are basic constructs for flow of control

**SEQUENCE** is a linear progression where one task is performed sequentially after another.   
**REPEATITION** is a loop with a simple conditional test at its beginning.   
**SELECTION** is a decision in which a choice is made between two alternative courses of action.

Keywords often used to indicate sequence, repetition and selection operations.

Sequence: requires no keyword, list action in the order they are to be performed.

Brush teeth

Wash face

Comb hair

Repetition: WHILE, FOR, REPEAT-UNTIL

Selection: IF-THEN-ELSE

Keywords often used to indicate common input, output, and processing operations.

Input: READ, GET   
Output: PRINT, DISPLAY  
Compute: COMPUTE, CALCULATE  
Initialize: SET   
Add one: INCREMENT

Here’s an example of pseudocode for the following requirement statement:

Given a radius, display the area of a circle

READ radius

COMPUTE area as radius times pi (3.14)

DISPLAY area

Here’s an example of pseudocode for the following requirement statement:

Given a radius, display the area of a circle. If the radius less than or equal to 0, display an error message, “Radius must be greater than 0”

READ radius

IF radius is less than or equal to 0 THEN

DISPLAY “Radius must be greater than 0”

ELSE

COMPUTE area as radius times pi (3.14)

DISPLAY area

Here’s an example of pseudocode for the following requirement statement:

Given a radius, display the area of a circle. If the radius less than or equal to 0, display an error message, “Radius must be greater than 0” and ask the user for another radius.

READ radius

WHILE radius is less than or equal to 0

DISPLAY “Radius must be greater than 0”

READ radius

COMPUTE area as radius times pi (3.14)

DISPLAY area