

Pseudocode: Purpose

- Develop an outline of the tasks required to solve a problem.
- Describe these tasks in prose (or flowcharts) without using specific programming language syntax.
- Save time by organize your thoughts before focusing on writing the code.
- Use psuedocode to guide code writing.

Pseudocode

General Process

- Understand the problem
- Decide what information is needed
- Identify the input(s) and output(s)
- Make a lists of the main tasks
- Break up tasks into small tasks
- Set up a logical structure to solve the problem
- *Detailed* pseudocode often corresponds line-to-line to finished code

Pseudocode

In pseudocode, do:

- Use general verbs (generate, compute, process, etc.) to describe the high-level tasks.
- Use specific verbs (initialize, set, reset, increment, compute, calculate, add, sum, multiply, print, display, input, output, edit, test, etc.) to describe low-level tasks.
- Use indentation to show logic.
- Describe loops using keywords souch as "DoWhile ... EndDo", "Do until ... EndDo", "Case ... EndCase", "If ... EndIf", etc.
- Write as a list of consecutive tasks (or, use a flowchart to indicate order of tasks).

Pseudocode

In pseudocode, do not:

- Declare variables
- Use code-specific keywords
- Use more than one line per statement

Pseudocode

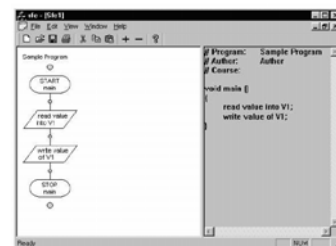
Simple example

- Problem statement: *Write a program that obtains two numbers from the user. Print out the sum of those numbers.*
- Pseudocode:
 - Prompt the user to enter the first number
 - Obtain the first number
 - Prompt the user to enter the second number
 - Obtain the second number
 - Add the first integer and second integer
 - Store the result in another variable
 - Display an output prompt that states the sum will be displayed
 - Display the results
- No single "right" answer!

Pseudocode

SFC - A Structured Flow Chart Editor

See: <http://linux.cs.sonoma.edu/~tiawatts/SFC/>



Pseudocode