Choosing Good Identifier Names

Many novice programmers believe that choosing good identifier names is not particularly important, but this is not true! Consider the following.

1) Failing to choose names carefully is often due to lack of attention to the algorithm which can lead to sloppy code.

2) Code written with poor names is unpleasant and difficult for persons to read (including your boss and/or your professor).

3) Code written with poor names is much more difficult to debug.

Selecting Good Variable Names

Since well chosen variable names can make a program much easier to read, you should invest the time needed to choose appropriate names. When possible, *use names that have mnemonic value*, i.e. that help the reader understand the intended meaning of the variable, constant, or function. Be specific rather then vague when choosing names. *Avoid nonstandard abbreviations* that will not be clear to others. Some examples of good and bad variable names are given below.

Bad Names	Good Names	Reason Name is Bad
`H`	`HourlyPayRate`	Name could stand for anything, hides purpose
`Found`	`DivisorFound`	Say what you found
`Table`	`PrimeTable`	Say what kind of table
`Value1`	`Smallest`	Be specific, why hide info from reader
`Count`	`DigitCount`	"Count" is the overused workhorse of many If you really are counting, say what it is
`Flag`	`InputErrorFlag`	"Flag" is rarely, if ever, a good name. If use it, then say what it is flagging
`Done`	`DoneSearching`	Should say what you're "Done" doing
`FrTbl`	`FrequencyTable`	Non-standard abbreviations should not be used

Sometimes, an identifier with mnemonic value will seem unnecessary or even inappropriate. For example, an int variable with no special qualities could appropriately be named "N".

Selecting Good Constant Names

Good programmers agree that constants such as "12" should be given a descriptive name, such as "NUMBERS_PER_ROW". Two reasons are given below. (Numbers, such as "12" are often referred to as literals.)

1) If constants are given descriptive names that capture the purpose of the constant, the program will be easier to understand.

2) Programs that use named constants are much easier to modify

It is a well established C and C++ tradition that named constants should have names consisting of all uppercase letters.

The underscore character should be used to separate the individual words that make up a constant identifier such as "NUMBERS_PER_ROW". Depending on how they are used, special constants, such as 0, 1, 10, etc., do not always have to be given descriptive names. For example, if you are dividing by 10 to eliminate the rightmost digit of an int, it is unnecessary and even unwise to give 10 a descriptive name.