

## Using Pseudocode and Flowcharts

- **Pseudocode** – An English-like representation of the logical steps it takes to solve a problem.
  - Code-like short statements
  - Independent of specific language syntax rules
  - Use indentation as in a programming language
- **Flowchart** – A graphical or pictorial representation of the logical steps it takes to solve a problem
  - Use standard symbols to represent input, processing, output and decisions
  - Draw flowlines to connect the steps and indicate order
  - As much as possible, flowchart should read top to bottom and left to right

### Example Pseudocode

```
start
    input myNumber
    set myAnswer = myNumber * 2
    output myAnswer
stop
```

- **myNumber** and **myAnswer** are **variables**
- a **Variable** is defined as a named memory location whose value can vary.

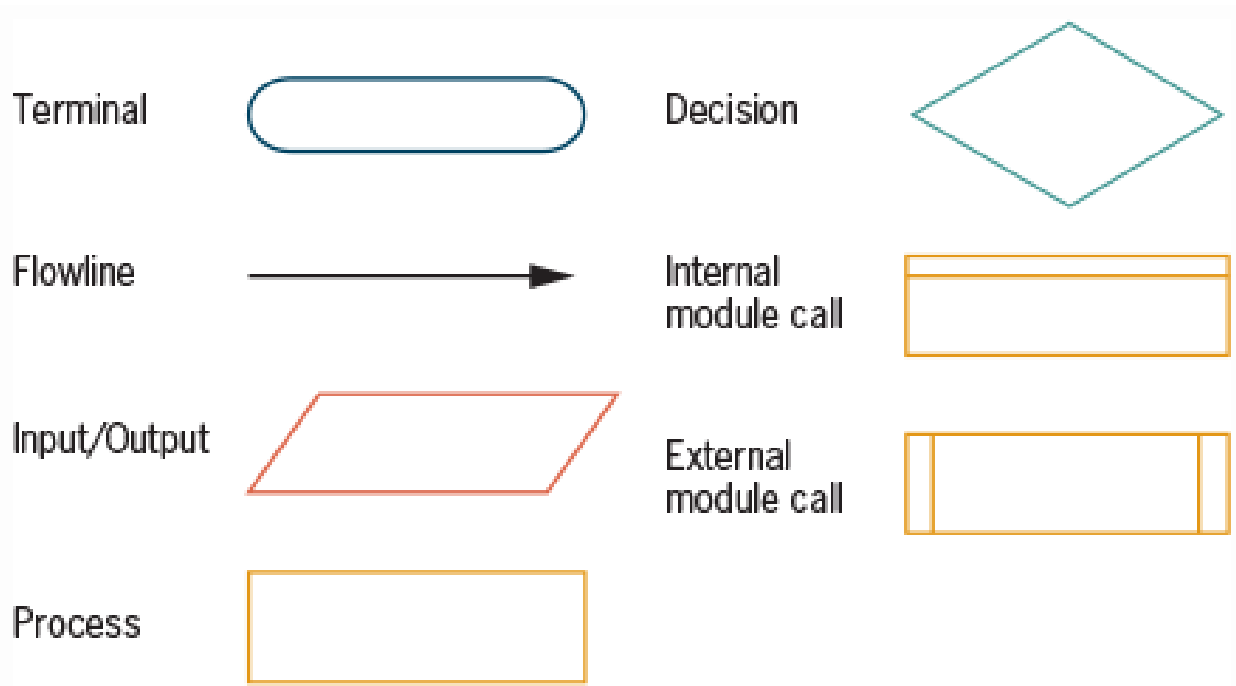
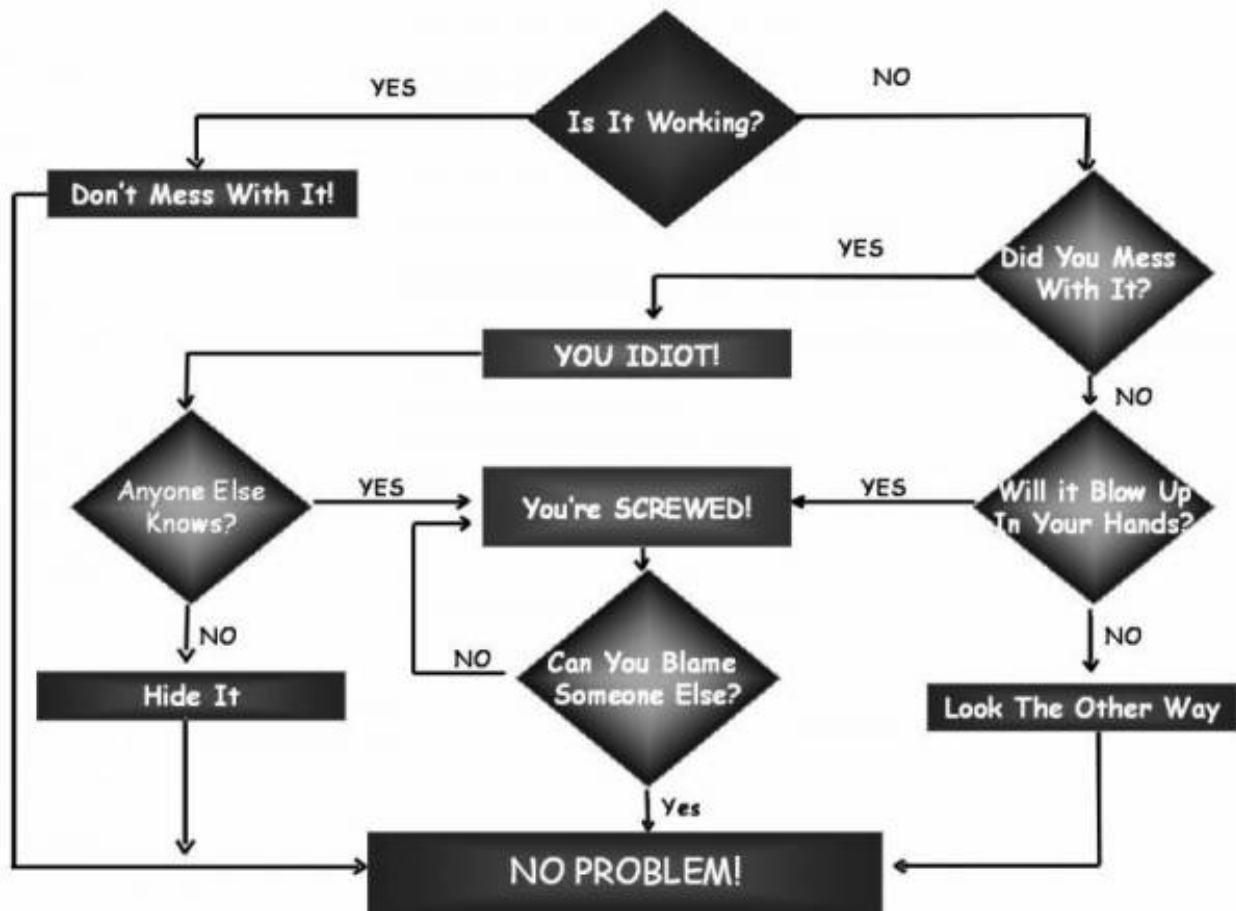
### Pseudocode Conventions

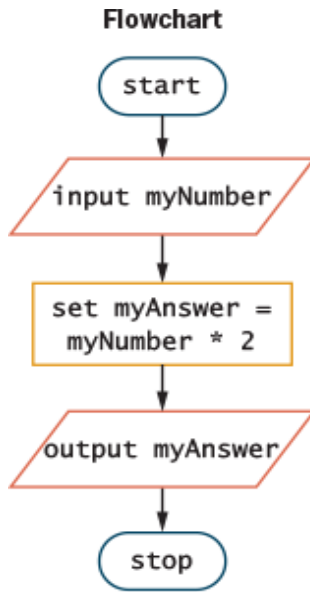
- Programs begin with **start** and end with **stop**; these two words are always aligned.
- Each program statement performs one action—for example, input, processing, or output.
- Each program statement appears on a single line if possible. When this is not possible, continuation lines are indented.
- Program statements are indented a few spaces more than **start** or the module name.
- Whenever a module name is used, it is followed by a set of parentheses.
- Modules begin with the module name and end with **return**. The module name and **return** are always aligned.
- Program statements begin with lowercase letters.
- No punctuation is used to end statements.

### Using Pseudocode

- No punctuation is used to end statements.
- May use **begin/end** rather than **start/stop**
- May use **get, input, read** for input
- May use **display, print, output, write** for output
- May write calculations as mathematical statements or English descriptions of the calculation, such as **double myNumber**

# Flowchart For Problem Resolution





**Pseudocode**

```

start
input myNumber
set myAnswer = myNumber * 2
output myAnswer
stop
  
```

Instead of  
start

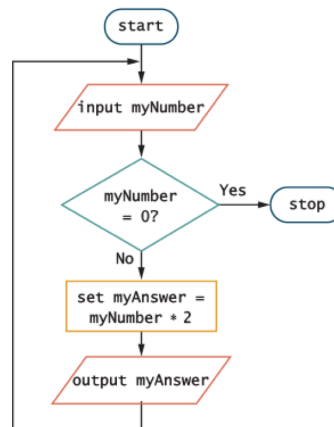
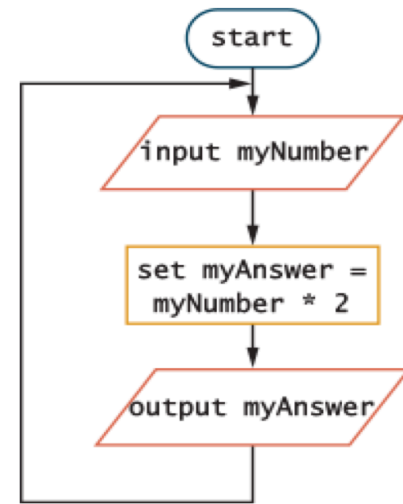
```

input myNumber
set myAnswer = myNumber * 2
output myAnswer
input myNumber
set myAnswer = myNumber * 2
output myAnswer
input myNumber
set myAnswer = myNumber * 2
output myAnswer
... (10,000 times!)
  
```

stop

**Infinite loop** – a repeating flow of logic with no end

Use a **sentinel value**, a value that the user enters that means to stop the program, or an **end-of-file condition** to terminate the loop.



Program makes a **decision**

- Draw a Flowchart and write Pseudocode to represent the logic of a program that computes the diameter and circumference of a circle. The program allows the user to enter a value for the radius. The program outputs the diameter and circumference.
- Modify the program logic to loop until the user enters a negative number for the radius