

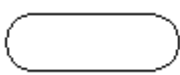

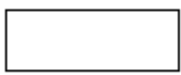
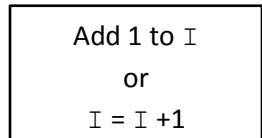

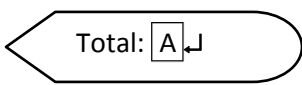

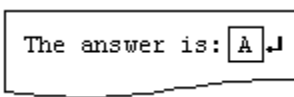
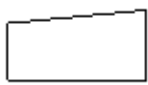
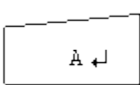
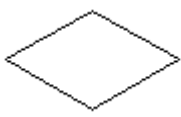
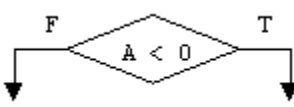

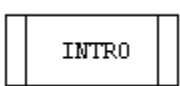

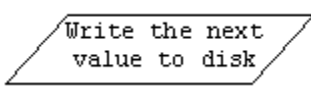
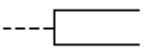
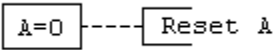

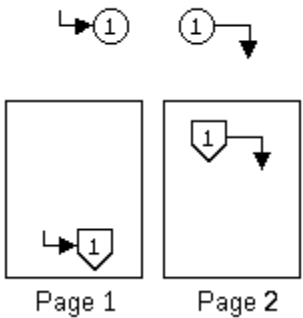


FLOWCHARTING SYMBOL TABLE

Note: In the examples below, assume that identifiers A and I represent integer variables.

SYMBOL / SHAPE	TYPE OF OPERATION / C++ Code Example	EXAMPLE OF SYMBOL USAGE
	Flow of Control Arrows indicating the sequence of steps ("flow of control").	
	Terminal activity - Start, Stop or End <code>{</code>	
	Assignment of a value to a variable, either directly or as the result of a calculation. <code>I = I + 1;</code>	
	Softcopy - screen output to a video display. <code>cout << "Total: " << A << endl;</code>	
	Hardcopy - document output to a printer. The C++ coding required to print hardcopy varies with each compiler.	
	Manual input from the keyboard to memory. <code>cin >> A;</code>	
	Decision - based on a relational condition, select from (branch to) multiple processes. <code>if (A < 0) {statements;}</code> <code>else {statements;}</code>	
	Sub-routine (a.k.a. "Function") used to indicate a process which is defined elsewhere. <code>INTRO (); /* Call Intro */</code>	
	General Input/Output of Data <code>/* Code depends on device involved */</code>	

	<p>Annotation for placing comments in logic.</p> <pre>A = 0; /* Reset A */</pre>	
	<p>Connectors: On-page (left) & Off-page (right).</p> <p>Used to either:</p> <ol style="list-style-type: none"> 1. Continue a flowchart at a different place either on or off the same piece of paper. 2. Close a selection branch or loop. 	 <p>Page 1 Page 2</p>

General Flowcharting Guidelines

- Symbols can be drawn any size (height or width) necessary to hold the symbol's contents.
- The shapes of most symbols imply the process. It is redundant to put the word "print" in a hardcopy symbol for example.
- Always put an arrowhead on each line connecting symbols to indicate the "flow of control". Do not assume that it will always be down or to the right.
- In output symbols, always differentiate between literal characters and identifiers by placing a box around the identifiers. For example in hardcopy output of variables named `DOLLARS` and `CENTS` mixed with some descriptive literal text, be sure to box the variable labels, like:

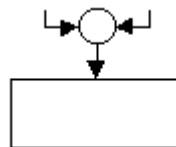
```
Total: DOLLARS dollars and CENTS cents. ↵
```

- Also note the inclusion of the symbol ↵ in the example above to indicate the use of a carriage return. It is also common to use the notation "<CR>" to represent a carriage return.
- The only symbols that may receive more than one incoming arrow are connectors. Never enter any other symbols using more than one arrow. If you want to do that, put a connector in front of the symbol and let the multiple arrows enter the connector.

Don't do this:



Do this:

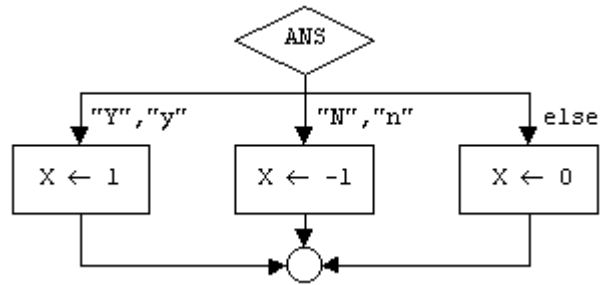


- The `switch` statement involves a special use of the diamond symbol. A flowchart for the following `switch` statement is shown to its right.

```

switch (ANS)
{
case 'Y':
case 'y': X = 1; break;
case 'N':
case 'n': X = -1; break;
default: X = 0; break;
}

```



Notice that the diamond contains only the name of the single variable to be evaluated. The "legs" that exit the decision diamond are each labeled with the unique values from a limited set of possible values for the variable ANS, including the "else" (default) option.