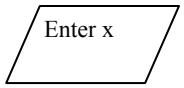
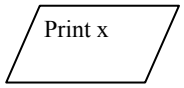
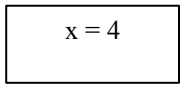
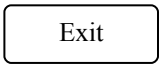
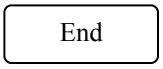
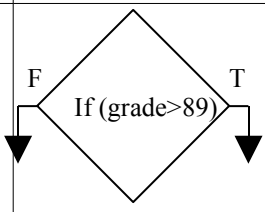
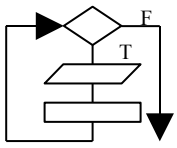
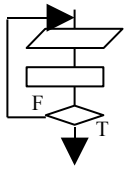
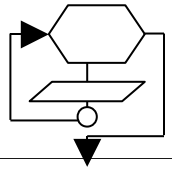
Pseudocode Reference Sheet

Command	Example	Action Performed	Flowchart Symbol	Corresponding C++ Command Example
Enter	Enter x	Get input from the keyboard. Assign the value of the input to the given variable (x in the example).		cin >> x;
Print	Print x Print “Hello”	Print output to the screen. Print the value of a variable or print a literal string of characters.		cout << x << “Hello”;
=	x = 4	Assign the value on the right-hand side to the variable on the left-hand side.		x = 4;
Exit	Exit	Exit the module. Return to the calling module.		
End	End	End the processes. (Use in the Control module)		

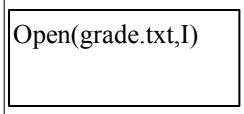
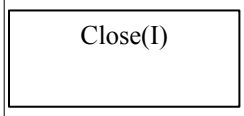
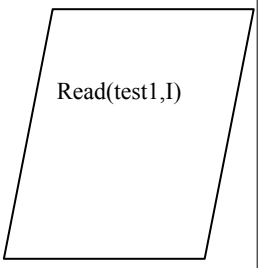
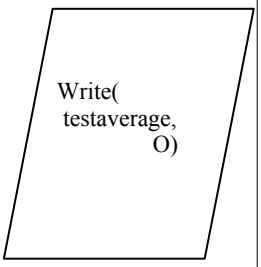
if (*cond*) then
  action
else
  action
	If (grade>89) then
	  print “A”
	else
	  print “Not an A”
		If the condition is true, execute the command under the else. If the condition is false, execute the command under the else.		if (grade > 89)
				  cout << “A”;
				else
				  cout << “Not an A”;

While/While-End	While (x < 100)
	  print x
	  x = x * 2
	While-End
		Execute the instructions inside the loop until the condition becomes false.		while (x < 100){
				  cout << x;
				  x = x*2;
				}

Repeat Until	Repeat
	  print x
	  x = x*2
	Until (x >=100)
		Execute the instructions inside the loop until the condition becomes true (false in a C++ do-while). The instructions will be executed at least once.		do {
				  cout << x;
				  x = x*2;
				}
				while (x < 100);

Automatic Counter	Loop: j = 1 to 5 by 1
	  print j
	Loop-End: j
		Execute the instructions inside the loop a set number of times.		for(j=0; j<5; j++){
				  cout << j;
				}

Pseudocode Reference Table (Continued)

Command	Example	Action Performed	Flowchart Symbol	Corresponding C++ Command Example
Open(*file, label*)	Open(grade.txt, I)	Opens the named file and uses the given label to refer to the file.		I.open("grade.txt"); Note: in C++, the variable (I in the example) must be declared for input (ifstream) or output (ofstream).
Close(*label*)	Close(I)	Closes the file referred to by the given label.		I.close( );
Read(*var*, *label*)	Read(test1, I)	Reads from the given input file (g in the example) until a space is encountered. The result is placed in the given variable (test1 in the example).		I.read(test, num); Reads num bytes from I into the variable test. I >> test; Reads from I until it gets to a space. The result is placed in the variable on the right-hand side (*test* in the example).
Write(*name*, *label*)	Write(testaverage, O)	Writes a literal string or the value of a variable (testaverage in the example) into the output file (O in the example).		O.write(testav, num); Writes num bytes of data from the variable testav into the output file, O. O << testav; Writes a literal string of characters or the value of a variable (testav in the example) into the output file (O in the example).
EOF(label)	EOF(I)	A logical function. TRUE if the end-of-file character has been reached in the input file (I in the example). FALSE otherwise.		I.eof( )

Relational Operators

Actions Performed: All operations return TRUE if the expression is true, and FALSE otherwise.
Use: These operators are used primarily in the conditions of if statements and loops.

Operator	Example	Corresponding C++ Operator
<	4 < 8 Result: TRUE	<
>	Age > 35 Result: TRUE if Age is greater than 35. FALSE otherwise	>
<=	Height <= 12 Result: TRUE if Height is less than or equal to 12. FALSE otherwise.	<=
>=	Height >= 12 Result: TRUE if Height is greater than or equal to 12. FALSE otherwise.	>=
== (Note: Don't confuse with =)	Age == 35 Result: TRUE if Age is equal to 35. FALSE otherwise.	==
<>	Age <> 35 Result: TRUE if Age is not equal to 35. FALSE otherwise.	!=

Logical Operators

Operator	Corresponding C++ Operator
AND	&&
OR	||
NOT	!

A	B	A AND B
T	T	T
T	F	F
F	T	F
F	F	F

A	B	A OR B
T	T	T
T	F	T
F	T	T
F	F	F

A	NOT A
T	F
F	T

Data Types

Data Type	Description	Example	Corresponding C++ Data Type
Character	A single ASCII character.	‘a’, ‘b’… ‘1’, ‘2’… ‘+’, ‘*’, etc.	char
Integer	A positive or negative whole number.	… -3, -2, -1, 0, 1, 2, 3…	int
Real	Any number on the number line.	0, 1.23423, 1/3, $\sqrt{2}$, π, etc.	float *or* double
String	A combination of characters.	“hello”, “12”, “Joe”, “Sue”, etc.	char[*number*] – an array of characters.
Input File	An input file stream		ifstream
Output File	An output file stream		ofstream