Computer Fundamentals

Dr. Safdar Nawaz Khan Marwat
DCSE, UET Peshawar

Lecture 21

safdar@uetpeshawar.edu.pk

Algorithm

- Derived from the name of a mathematician
  - Mohammed ibn-Musa al-Khwarizmi
- Set of steps to accomplish a task
  - Always leads to a solution

- Before writing a program to solve a problem, we require
  - Thorough understanding of problem
  - Carefully planned approach for solving
  - To understand available building blocks
  - Employ proven program construction techniques

- Important aspects of algorithm
  - Actions to execute
  - Order of execution

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Algorithm (cont.)

- Consider the rise-and-shine algorithm
  - Get out of bed
  - Take a shower
  - Get dressed
  - Eat breakfast
  - Go to University

- Suppose same steps are performed in different order
  - Get out of bed
  - Get dressed
  - Take a shower
  - Eat breakfast
  - Go to University

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Pseudocode

- Fake code
- Artificial and informal language
  - To develop algorithms without details of language syntax
  - Similar to everyday English
  - Convenient and user friendly
  - Not actual computer programming language
- Used to conceptualize program before implementation

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Pseudocode (cont.)

- Example of adding two numbers

```
#include <iostream>
using namespace std;

int main ()
{
  int num1, num2, result;
  cout <<"Enter 1st number: ";
  cin >> num1;
  cout <<"Enter 2nd number: ";
  cin >> num2;
  result = num1 + num2;
  cout << "\nSum = " << result;
  cout << endl;
  return 0;
}
```

Prompt the user to enter integer num1
READ num1

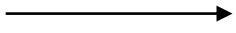
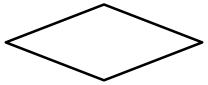
Prompt the user to enter integer num2
READ num2

SET integer result as sum of num1 and num2
DISPLAY result

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Flowchart

- Diagrammatic representation of algorithm
- Helpful in explaining program to others

Symbol	Purpose	Description
	Flow line	Indicate flow of logic by connecting symbols
	Terminal (Start/Stop)	Represent start and end of flowchart
	Input/Output	Input and output operation
	Processing	Arithmetic operations and data-manipulations
	Decision	Represent operation with alternatives
	Predefined function	Represent group of statements performing one task

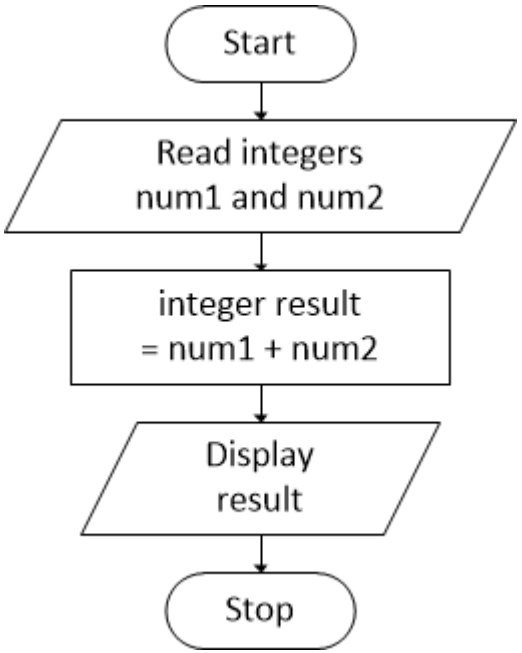
Source: *http://www.programiz.com/article/flowchart-programming*

Flowchart (cont.)

- Example of adding two numbers

```
#include <iostream>
using namespace std;

int main ()
{
  int num1, num2, result;
  cout <<"Enter 1st number: ";
  cin >> num1;
  cout <<"Enter 2nd number: ";
  cin >> num2;
  result = num1 + num2;
  cout << "\nSum = " << result;
  cout << endl;
  return 0;
}
```



Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Arithmetic Operations

- Arithmetic operations very important in programming
- Several operations possible

Operation	Arithmetic operator	Algebraic expression	Programming expression	Precedence
Addition	+	$d + 7$	`d + 7`	Third
Subtraction	-	$a - c$	`a - c`	Third
Multiplication	*	$b\ t$ or $b \cdot t$	`b * t`	Second
Division	/	x / y or $x \div y$	`x / y`	Second
Modulus	%	$n \% d$	`n % d`	Second
Parentheses	( )	( )	`( )`	First

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

Integer Variable

- Integer variable hold value to the left of decimal point
- Values to the right of decimal point discard

```
SET integer i equal to 3.9
DISPLAY i
```

- The output would be 3 and not 3.9
- Integer does not hold value to the right of decimal point

if statement

- Allows program to take alternative action
  - Based on whether a condition is true or false
- If condition is true
  - Statement in body of `if` statement executed
- If condition is false
  - Body statement not executed
- Equality operators and relational operators used

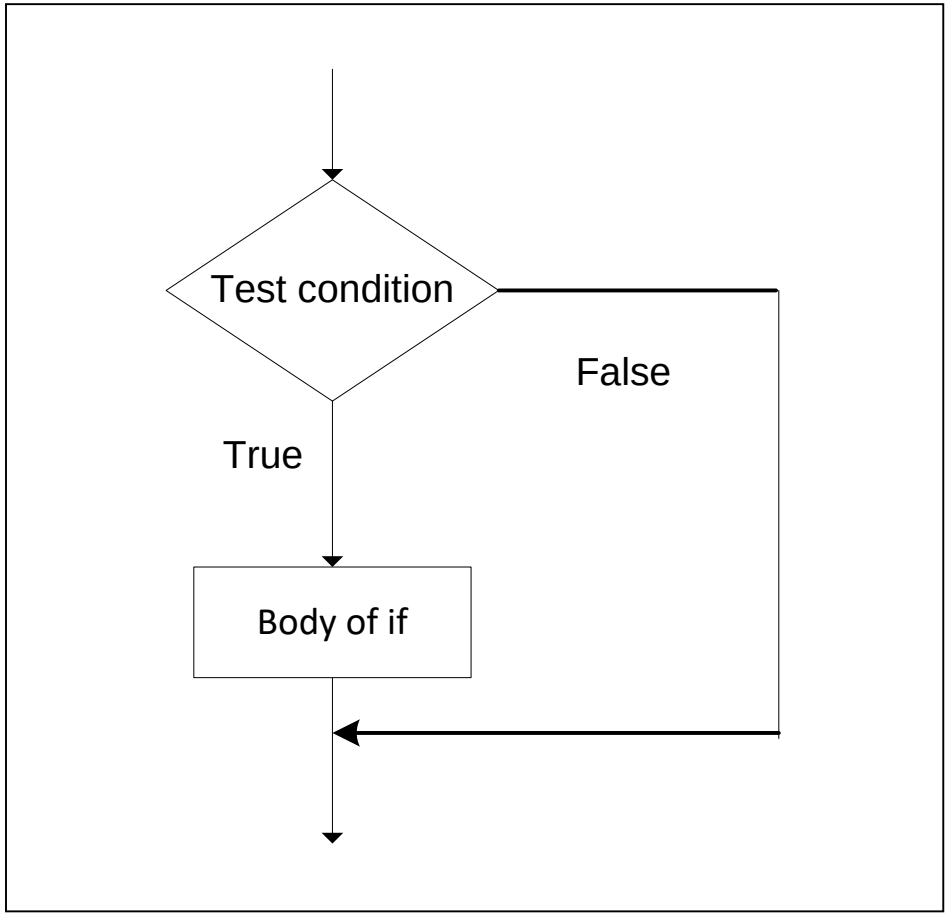
Operator	Equality or relational operator	Sample condition	Meaning of condition
Relational Operators	>	`x > y`	x greater than y
	<	`x < y`	x less than y
	>=	`x >= y`	x greater than or equal to y
	<=	`x <= y`	x less than or equal to y
Equality operators	==	`x == y`	x equal to y
	!=	`x != y`	x not equal to y

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

if statement (cont.)

- Pseudocode and flow chart of `if` statement

```
IF condition
   Body of IF
END IF
```

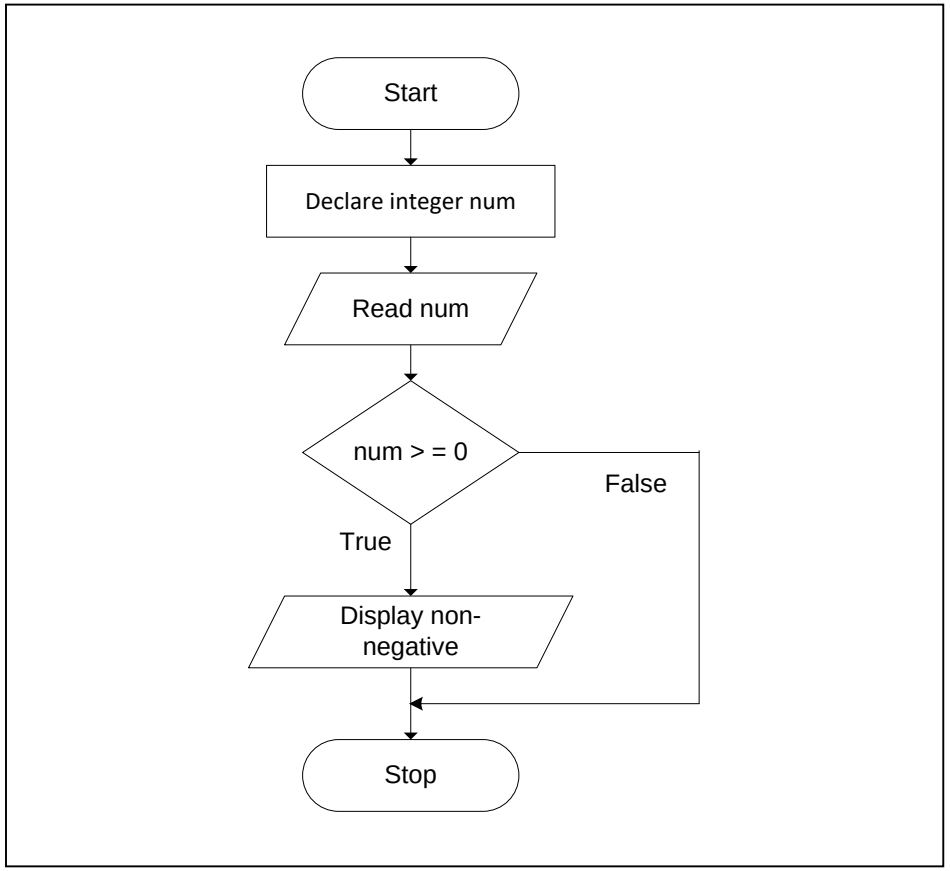


if statement (cont.)

- A program to check whether an integer entered by user is non-negative

```
Prompt the user to enter integer num
READ num

IF num is greater than or equal to 0
   DISPLAY non-negative
END IF
```



if else statement

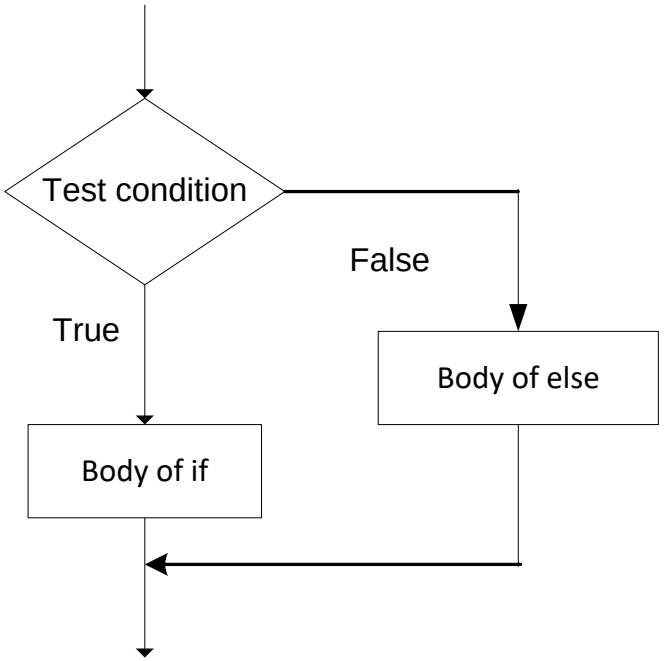
- Allows program to take alternative action
  - Based on whether a condition is true or false
- If condition is true
  - Statement in body of `if` statement executed
- If condition is false
  - Body statement of `else` statement executed
- Equality operators and relational operators used

Source: Deitel & Deitel, "*C++, How to Program*", 8th Edition

if else statement (cont.)

- Pseudocode and flow chart of `if else` statement

```
IF condition
   Body of IF
ELSE
   Body of ELSE
END IF
```



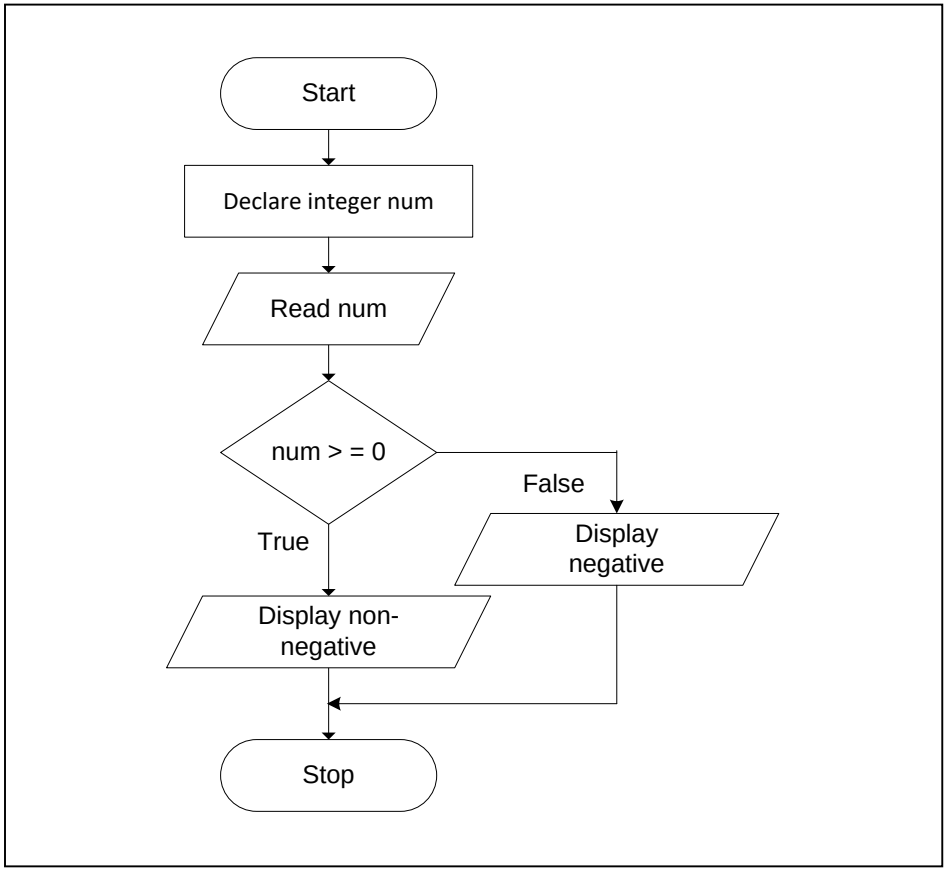
Source: *http://www.programiz.com/c-programming/c-if-else-statement*

if else statement (cont.)

- A program to check whether an integer entered by user is non-negative or negative

```
Prompt the user to enter integer num
READ num

IF num is greater than or equal to 0
   DISPLAY non-negative
ELSE
   DISPLAY negative
END IF
```



Source: *http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html*

Nested if else statements

- `if else` statements nested inside another `else`
- If one condition fails, another condition is checked
- Used when more than one test conditions are required
  - If first condition is true, body executed
  - Otherwise, second condition is tested
    - If second condition is true, body executed
    - Otherwise, third condition is tested

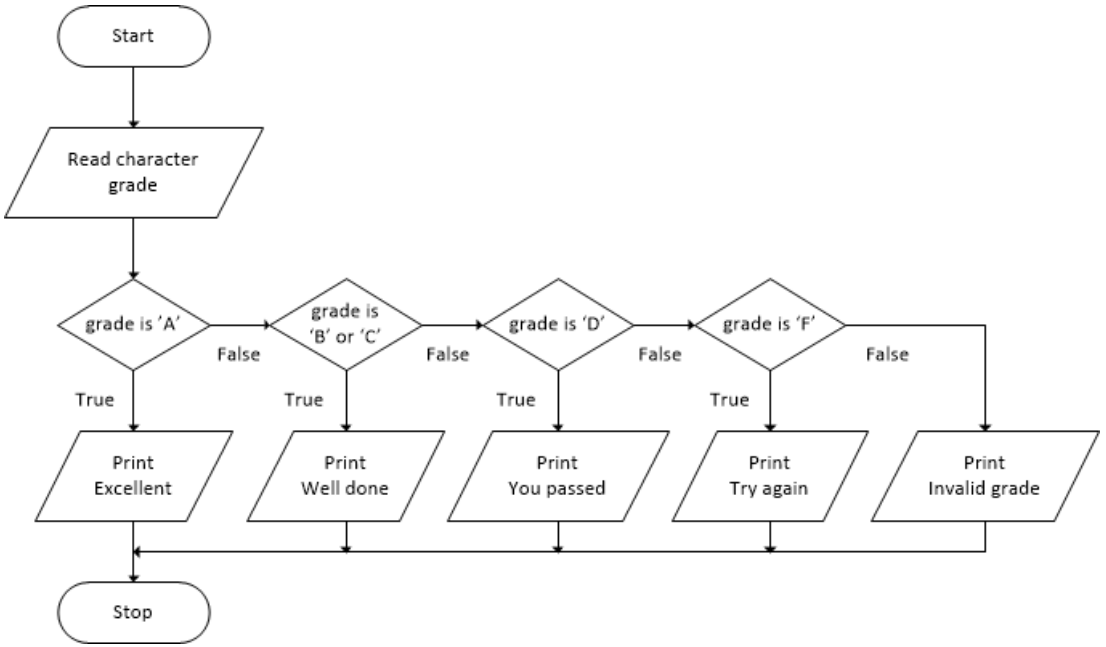
Source: *http://www.programiz.com/c-programming/c-if-else-statement*

Nested if else statements (cont.)

- A program about exam grade (A, B, C, D, F) and remarks

```
Prompt the user to enter character grade
READ grade

IF grade is ‘A’
   DISPLAY Excellent
ELSE
   IF grade is ‘B’ or ‘C’
      DISPLAY Well done
   ELSE
      IF grade is ‘D’
         DISPLAY You passed
      ELSE
         IF grade is ‘F’
            DISPLAY Try again
         ELSE
            DISPLAY Invalid grade
         END IF
      END IF
   END IF
END IF
```



Class Task

- Write the pseudocode and draw flowchart for a program that reads an integer from the user and checks whether the number is odd or even.