C Programming for Engineers

Structured Program

UNIVERSITY AT ALBANY
State University of New York

ICEN 360– Spring 2017

Prof. Dola Saha

Steps to writing a program

- Understand the problem
- Plan a solution
  - Step by step procedure

Algorithm

- The solution to any computing problem involves executing a series of actions in a specific order.
- A procedure for solving a problem in terms of
  - the actions to be executed, and
  - the order in which these actions are to be executed
- is called an algorithm.
- Correctly specifying the order in which the actions are to be executed is important.

Order matters

- Example “rise-and-shine” algorithm

In-Order	
1. Get out of bed	
2. Take of pajamas	
3. Take a shower	
4. Get dressed	
5. Eat breakfast	
6. Carpool to work	

Order matters

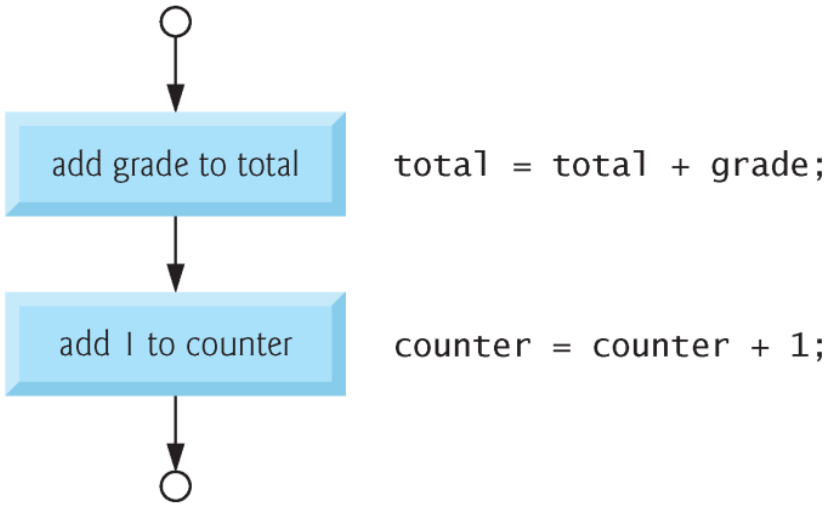
- Example "rise-and-shine" algorithm

In-Order	Out-of-Order
1. Get out of bed	1. Get out of bed
2. Take of pajamas	2. Take of pajamas
3. Take a shower	3. Get dressed
4. Get dressed	4. Take a shower
5. Eat breakfast	5. Eat breakfast
6. Carpool to work	6. Carpool to work

- Specifying the order in which statements are to be executed in a computer program is called program control.

Flow Chart

- Graphical representation of an algorithm
- Uses certain special-purpose symbols such as rectangles, diamonds, rounded rectangles, and small circles
- Symbols are connected by arrows called flowlines



Flow Chart

- Rectangle symbol or action symbol indicate any type of action including a calculation or an input/output operation.
- The flowlines indicate the order in which the actions are performed.

Pseudocode

- Artificial, informal, user-friendly, convenient, English-like
- They are NOT executed on computers
- Can be easily converted into ANY programming language
- Consists of actions and decision statements

```
1   Set total to zero
2   Set grade counter to one
3
4   While grade counter is less than or equal to ten
5       Input the next grade
6       Add the grade into the total
7       Add one to the grade counter
8
9   Set the class average to the total divided by ten
10  Print the class average
```

Control Structures

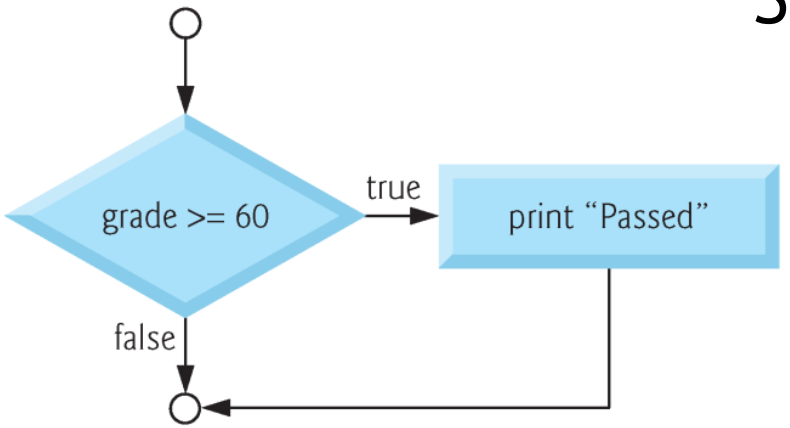
- Sequential execution: statements are executed one after another
- Transfer of control: Some C statements can specify that next statement to be executed MAY NOT be the next statement

Selection Statement

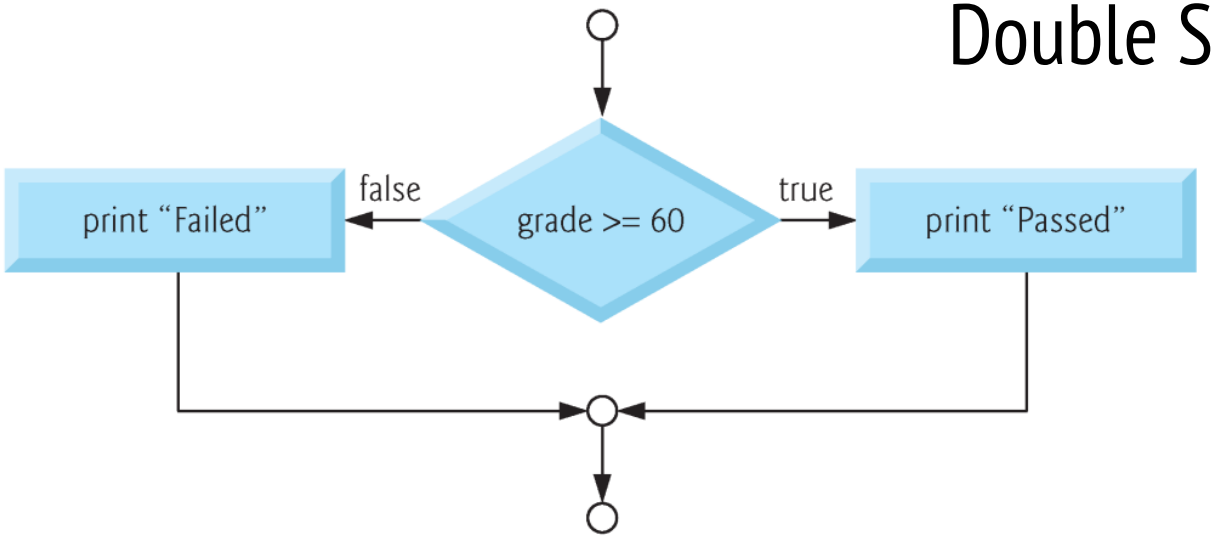
- `If` Statement
  - `If` :
    - Performs a set of actions if condition is TRUE,
    - otherwise skip
  - `If … else` :
    - Performs a set of actions if condition is TRUE,
    - otherwise performs a different set of actions
- `Switch` Statement:
  - Performs one of many different set of actions
- Used to choose among alternative courses of action.

Selection Statement in Flow Chart

Single Selection



Double Selection



Selection Statement in Pseudocode

If student's grade is greater than or equal to 60
    Print "Passed"

If student's grade is greater than or equal to 60
    Print "Passed"
else
    Print "Failed"

Selection Statement in C

```
if ( grade >= 60 ) {
   printf( "Passed\n" );
} // end if
```

```
if ( grade >= 60 ) {
   printf( "Passed\n" );
} // end if
else {
   printf( "Failed\n" );
} // end else
```

Example: Swap values of two variables

```
1.  if (x > y) {                    /* Switch x and y */
2.        temp = x;                 /* Store old x in temp */
3.        x = y;                    /* Store old y in x */
4.        y = temp;                 /* Store old x in y */
5.  }
```

Conditional Operator (?)

- C's only ternary operator—it takes *three* operands.
- The first operand is a *condition*.
- The second operand is the value for the entire conditional expression if the condition is *TRUE.*
- The third operand is the value for the entire conditional expression if the condition is *FALSE*.
- Example:
  - `printf( grade >= 60 ? "Passed" : "Failed" );`

Classroom Assignment

- Develop an algorithm to find a number is odd or even
- Write a pseudocode to check if a number is odd or even
- Write a C code that takes an integer as input from the user and prints out whether it is odd or even number

Nested if… else Statements

```
if ( grade >= 90 )
   puts( "A" );
else
   if ( grade >= 80 )
      puts("B");
   else
      if ( grade >= 70 )
         puts("C");
      else
         if ( grade >= 60 )
            puts( "D" );
         else
            puts( "F" );
```

If … else if Statement

```
if ( grade >= 90 )
   puts( "A" );
else if ( grade >= 80 )
   puts( "B" );
else if ( grade >= 70 )
   puts( "C" );
else if ( grade >= 60 )
   puts( "D" );
else
   puts( "F" );
```

Compound Statement

- The `if` selection statement expects only one statement in its body
- To include several statements in the body of an `if`, the set of statements are included in braces

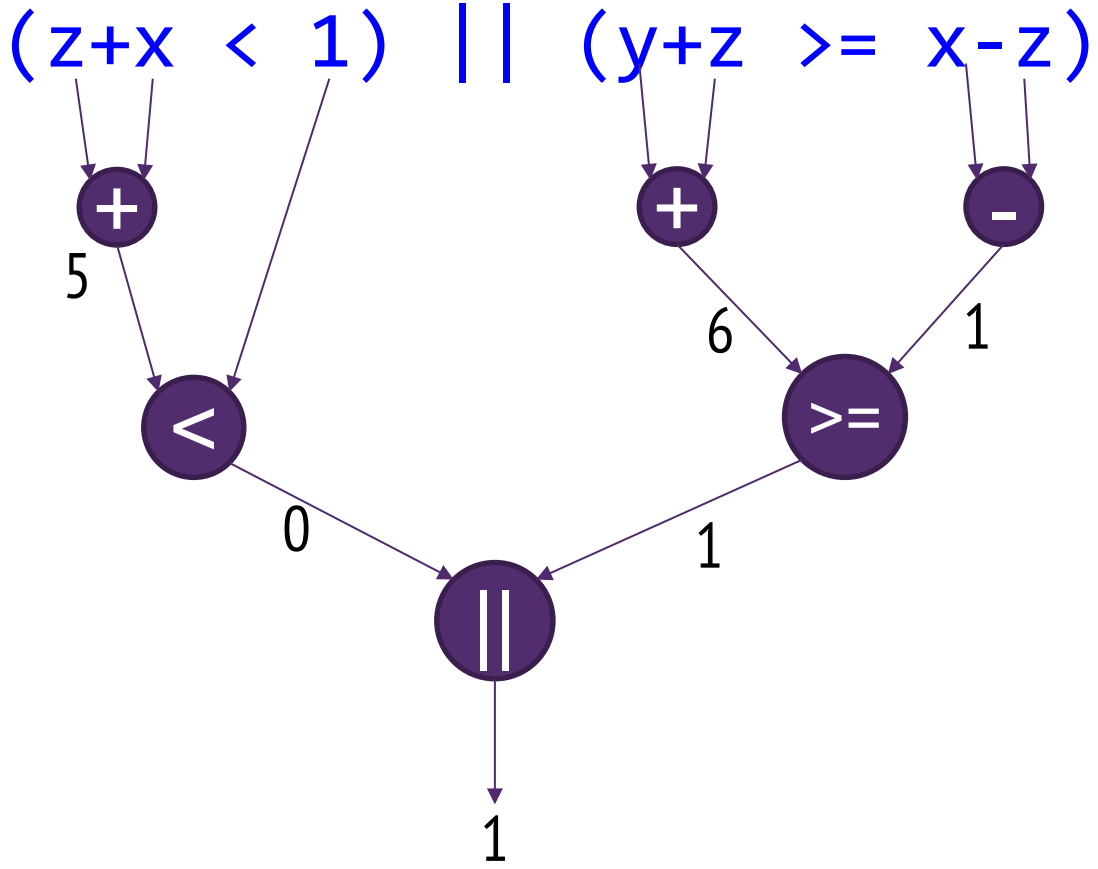
```
if ( grade >= 60 )
   puts( "Passed. " );
// end if
else {
   puts( "Failed. " );
   puts( ”Take this course again. " );
} // end else
```

```
{
    statement;
    statement;
        .
        .
        .
    statement;
}
```

Precedence

Operator	Precedence
()	highest (evaluated first)
* / %	
+ -	
< <= >= >	
== !=	
&&	
||	
=	lowest (evaluated last)

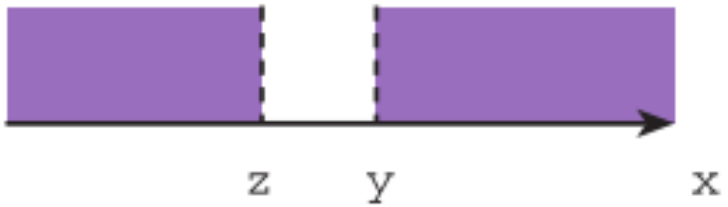
Condition Statements



Common usage in program

- Check range of x


```
x >= min  &&  x <= max
```




```
x < z || x > y
```

English to C Logical Expression

English Condition
`x` and `y` are greater than `z`
`x` is equal to `1.0` or `3.0`
`x` is in the range `z` to `y`, inclusive
`x` is outside the range `z` to `y`

English to C Logical Expression

English Condition	Logical Expression	Evaluation
`x` and `y` are greater than `z`	`x > z && y > z`	`1 &&` 1 is 1 (true)
`x` is equal to `1.0` or `3.0`	`x == 1.0 || x == 3.0`	`0 ||` 1 is 1 (true)
`x` is in the range `z` to `y`, inclusive	`z <= x && x <= y`	`1 &&` 1 is 1 (true)
`x` is outside the range `z` to `y`	`!(z <= x && x <= y)` `z > x || x > y`	`!(1 && 1)` is 0 (false) `0 ||` 0 is 0 (false)