Selection Structures: if and switch Statements

Chapter 4

Problem Solving & Program Design in C

Eighth Edition

Jeri R. Hanly & Elliot B. Koffman

Chapter Objectives

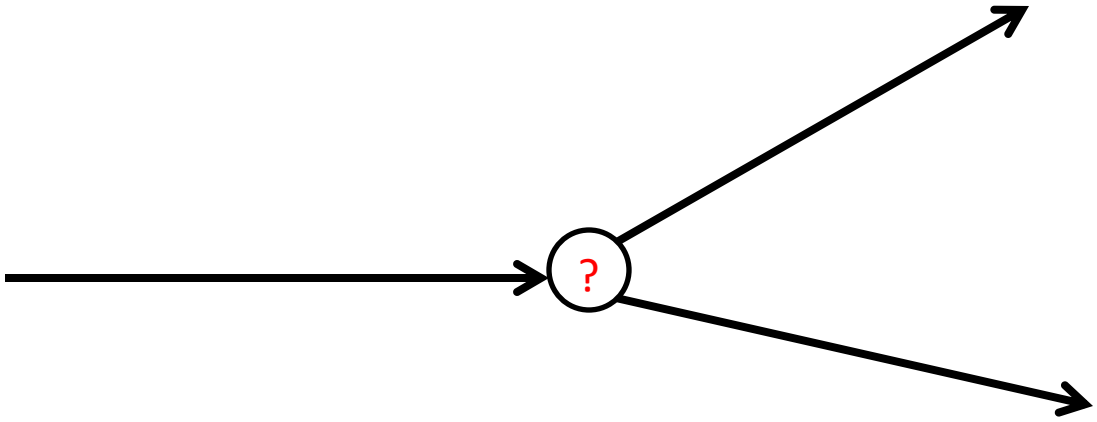
- To become familiar with the three kinds of control structures: sequence, selection, and repetition
- To understand compound statements
- To learn how to compare numbers and characters
- To learn how to use the relational, equality, and logical operators to write expressions that are true or false

Chapter Objectives

- To learn how to write selection statements that choose between two alternatives in a program using the if statement
- To learn how to implement decisions tin algorithms using the if statement
- To understand how to select among more than two alternatives by nesting if statements
- To learn how to use the switch statement as another technique for selecting among multiple alternatives

Control Structures

- selection control structure
  - a control structure that chooses among alternative program statements



Conditions

- an expression that is either false
  - represented by 0
- or true
  - usually represented by 1

```
rest_heart_rate  >  75
```

Relational and Equality Operators

Operator	Meaning	Type
<	less than	relational
>	greater than	relational
<=	less than or equal to	relational
>=	greater than or equal to	relational
==	equal to	relational
!=	not equal to	equality

Logical Operators

- logical expressions
  - an expression that uses one or more of the logical operators
    - && (and)
    - || (or)
    - ! (not)

Logical Operators

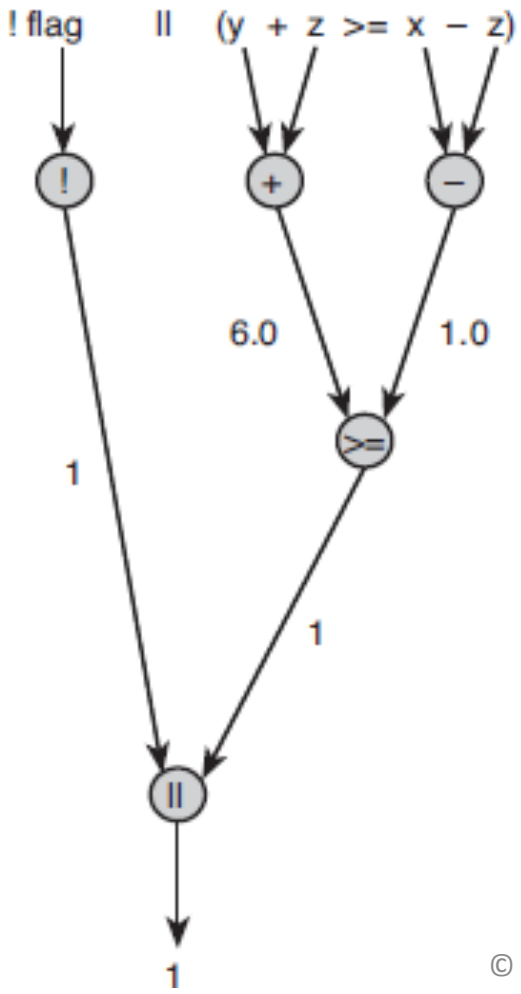
- logical complement (negation)
  - the complement of a condition had the value 1 (true) when the condition's value is 0 (false)
  - the complement of a condition has the value 0 (false) when the condition's value is nonzero (true)

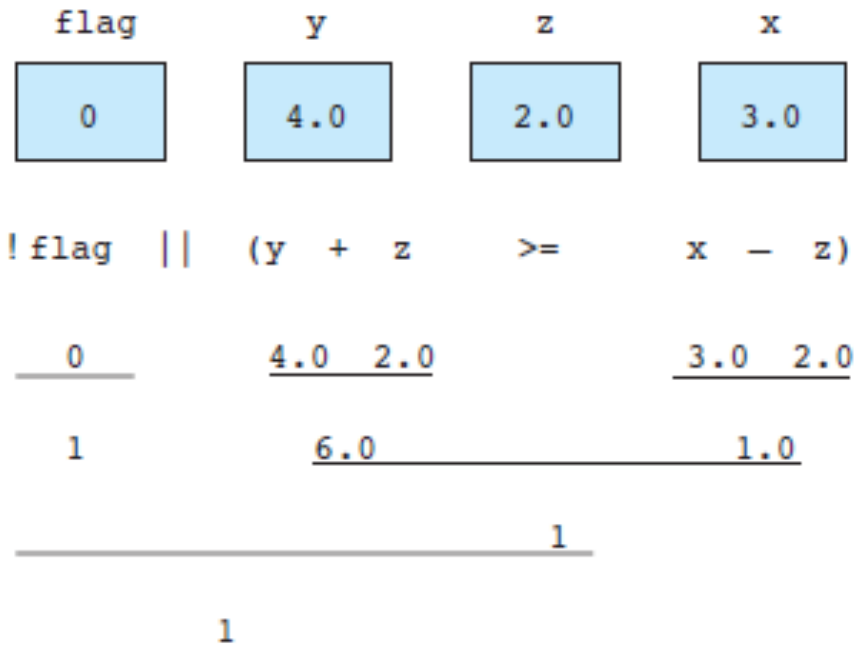
```
! (0 <= n && n <= 100)
```

Operator Precedence

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

Figure 4.1
Evaluation Tree and Step-by-Step Evaluation for !flag || (y + z >= x - z)





Short-Circuit Evaluation

- stopping evaluation of a logical expression as soon as its value can be determined

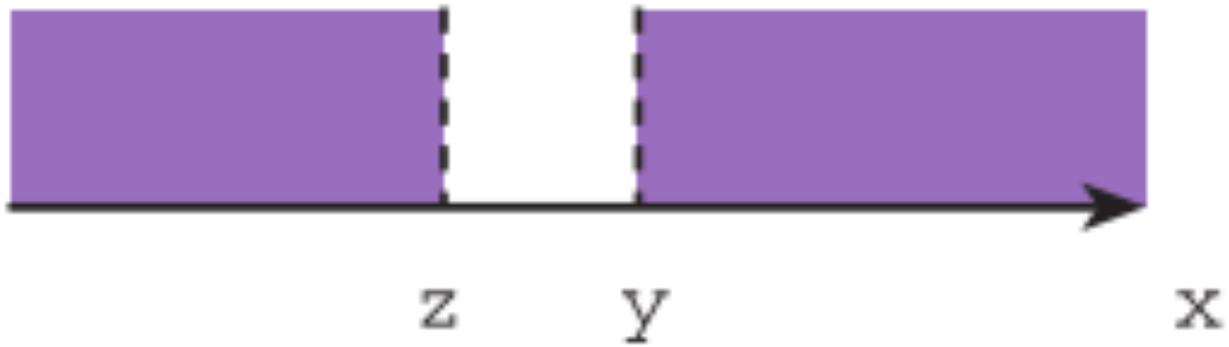
```
(div != 0 && (num % div == 0))
```

Figure 4.2

Range of True Values for min <= x && x <= max

Figure 4.3

Range of True Values for
z > x || x > y



Comparing Characters

Expression	Value
'9' >= '0'	1 (true)
'a' < 'e'	1 (true)
'B' <= 'A'	0 (false)
'Z' == 'z'	0 (false)
'a' <= 'A'	System dependent
'a' <= ch && ch <= 'z'	1 (true) if ch is a lowercase letter

The if-statement

making decisions

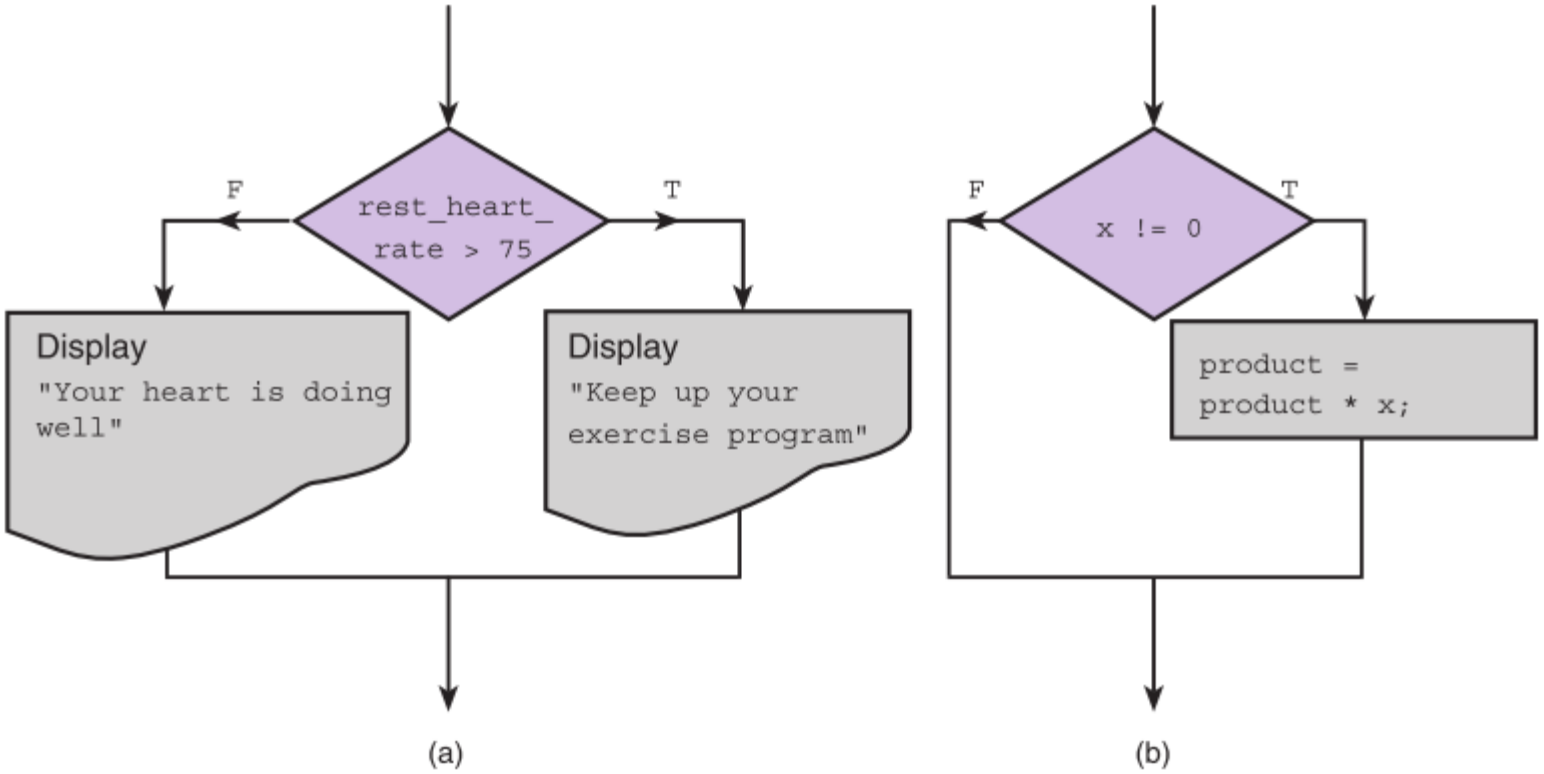
if-statement with one alternative

```
if  (x != 0)
        product = product * x;
```

Compound Statement

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

Figure 4.4
Flowcharts of if Statements with (a) Two Alternatives and (b) One Alternative



if-statement with two alternatives

```
if  (rest_heart_rate > 75)
        printf("Keep up your exercise program!\n");
else
        printf("Your hear is doing well!\n");
```

Let's finish off a C program using this if statement

The switch statement

- also used to select one of several alternatives
- useful when the selection is based on the value of
  - a single variable
  - or a simple expression
- values may of type int or char
  - not double

Syntax

```
switch (controlling expression) {
        label set1
                statements1
                break;
        label set2
                statements2
                break;
                .
                .
                .
        label setn
                statementsn
                break;
```

Let's look at ships.c and add a new label set

Wrap Up

- Use control structures to control the flow of statement execution in a program.
- Use selection control structures to represent decisions in an algorithm.
- Nested if statements are common in C and are used to represent decisions with multiple alternatives.
- The switch statement implements decisions with several alternatives where the alternative selected depends on the value of a variable or (controlling) expression.