

# Selection Structures: if and switch Statements

## Chapter 4

*Problem Solving & Program Design in C*

*Eighth Edition*

*Jeri R. Hanly & Elliot B. Koffman*

# Chapter Objectives

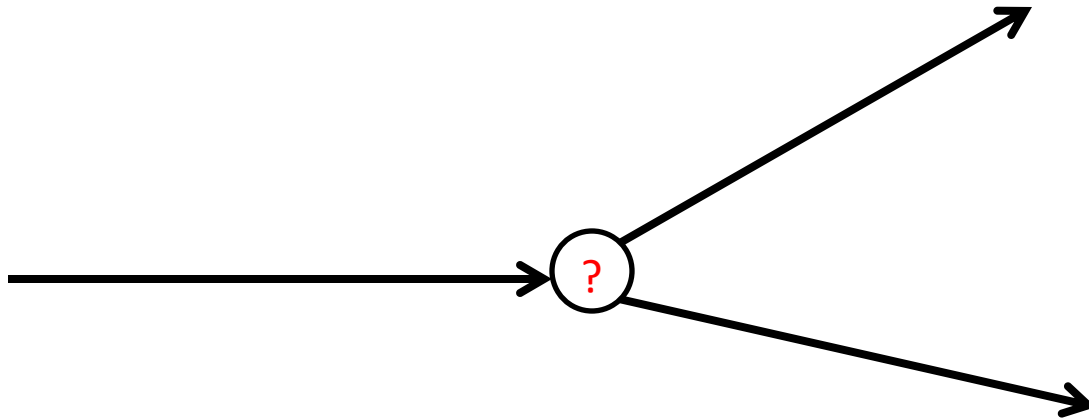
- To become familiar with the three kinds of control structures: sequence, selection, and repetition
- To understand compound statements
- To learn how to compare numbers and characters
- To learn how to use the relational, equality, and logical operators to write expressions that are true or false

# 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 in 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	equality
!=	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 4.2

Range of True Values for  
 $\text{min} \leq x \ \&\& \ x \leq \text{max}$



Figure 4.3

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

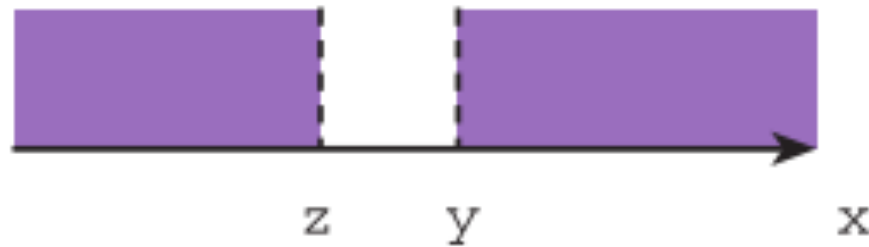
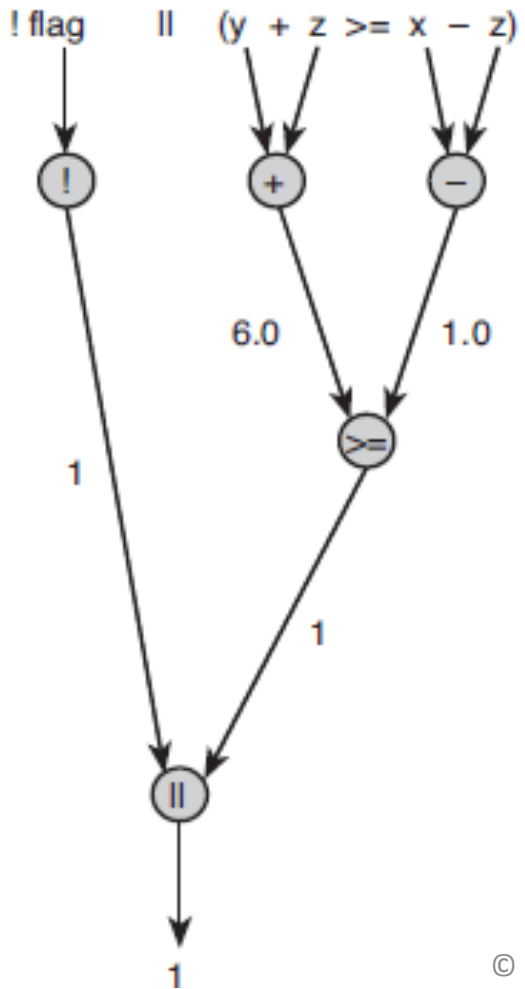


Figure 4.1

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



flag	y	z	x
0	4.0	2.0	3.0

! flag		(y + z	>=	x - z)
0		4.0 2.0		3.0 2.0
1		6.0		1.0
				1
	1			

# Short-Circuit Evaluation

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

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

# Comparing Characters

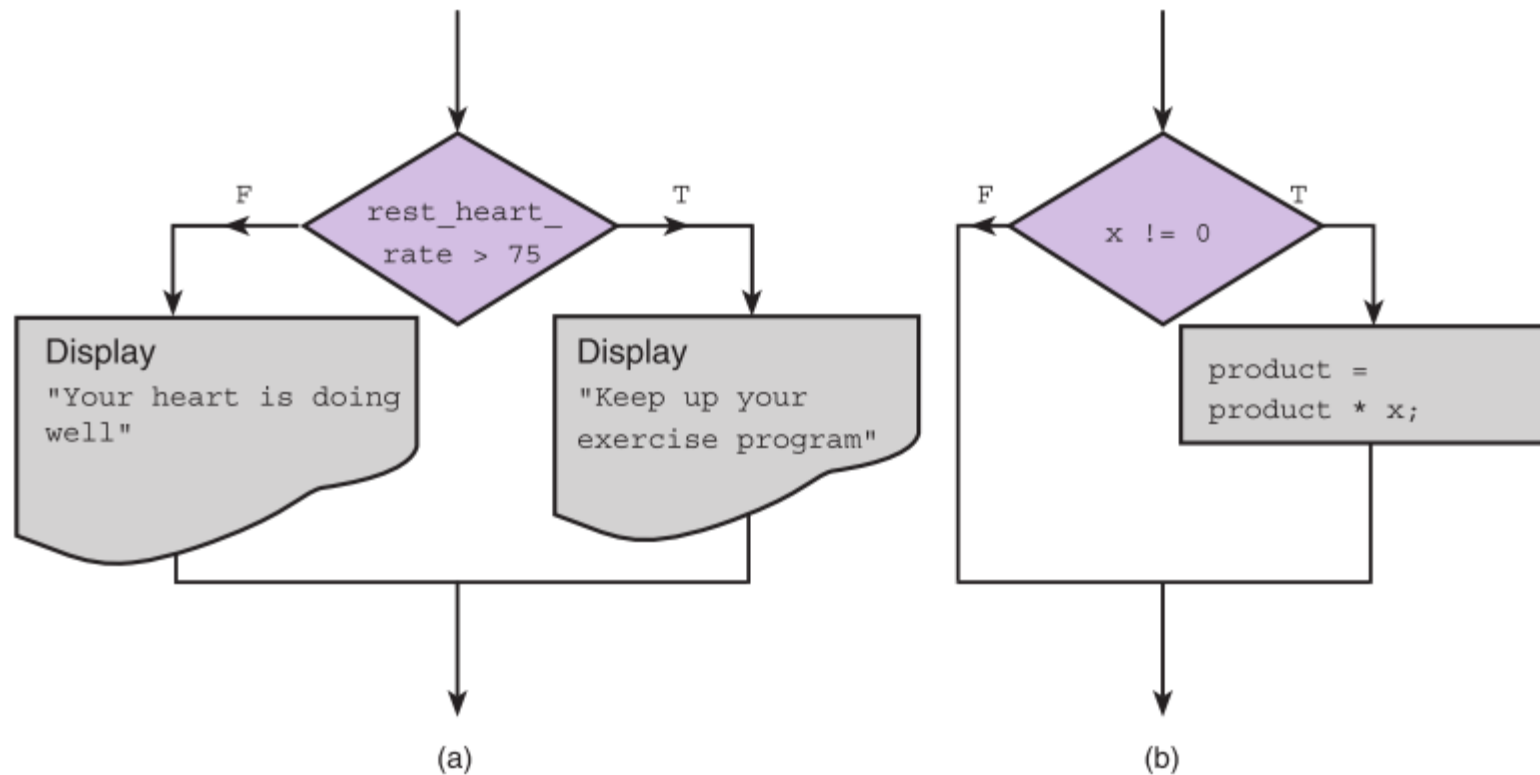
Expression	Value
'9' >= '0'	1 (true)
'a' < 'e'	1 (true)
'B' <= 'A'	0 (false)
'Z' == 'z'	0 (false)
'a' <= 'A'	System dependent
'a' <= ch && ch <= 'z'	1 (true) if ch is a lowercase letter

# The if-statement

making decisions

# Figure 4.4

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





# if-statement with one alternative

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

# if-statement with two alternatives

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

# Figure 4.5 Program Using an *if* statement for selection

```
1. /*
2.  * Displays message about heart rate.
3.  */
4. #include <stdio.h>
5.
6. int main(void)
7. {
8.     int pulse;           /* resting pulse rate for 10 secs */
9.     int rest_heart_rate; /* resting heart rate for 1 minute */
10.
11.     /* Enter your resting pulse rate */
12.     printf("Take your resting pulse for 10 seconds.\n");
13.     printf("Enter your pulse rate and press return> ");
14.     scanf("%d", &pulse);
15.
16.     /* Calculate resting heart rate for minute */
17.     rest_heart_rate = pulse * 6;
18.     printf("Your resting heart rate is %d.\n", rest_heart_rate);
19.
20.     /* Display message based on resting heart rate */
21.     if (rest_heart_rate > 56)
22.         printf("Keep up your exercise program!\n");
23.     else
24.         printf("Your heart is in excellent health!\n");
25.
26.     return (0);
27. }
```

## Sample Run 1

```
Take your resting pulse for 10 seconds.
Enter your pulse rate and press return> 12
Your resting heart rate is 72.
Keep up your exercise program!
```

## Sample Run 2

```
Take your resting pulse for 10 seconds.
Enter your pulse rate and press return> 9
Your resting heart rate is 54.
Your heart is in excellent health!
```

# Figure 4.6

## if Statement to Order $x$ and $y$

```
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. }
```

# Nested if-statement

- an if statement with another if statement as its true task or its false task

```
if (x > 0)
    num_pos = num_pos + 1
else
    if (x < 0)
        num_neg = num_neg + 1
    else /* x equals 0 */
        num_zero = num_zero + 1
```

# Figure 4.11

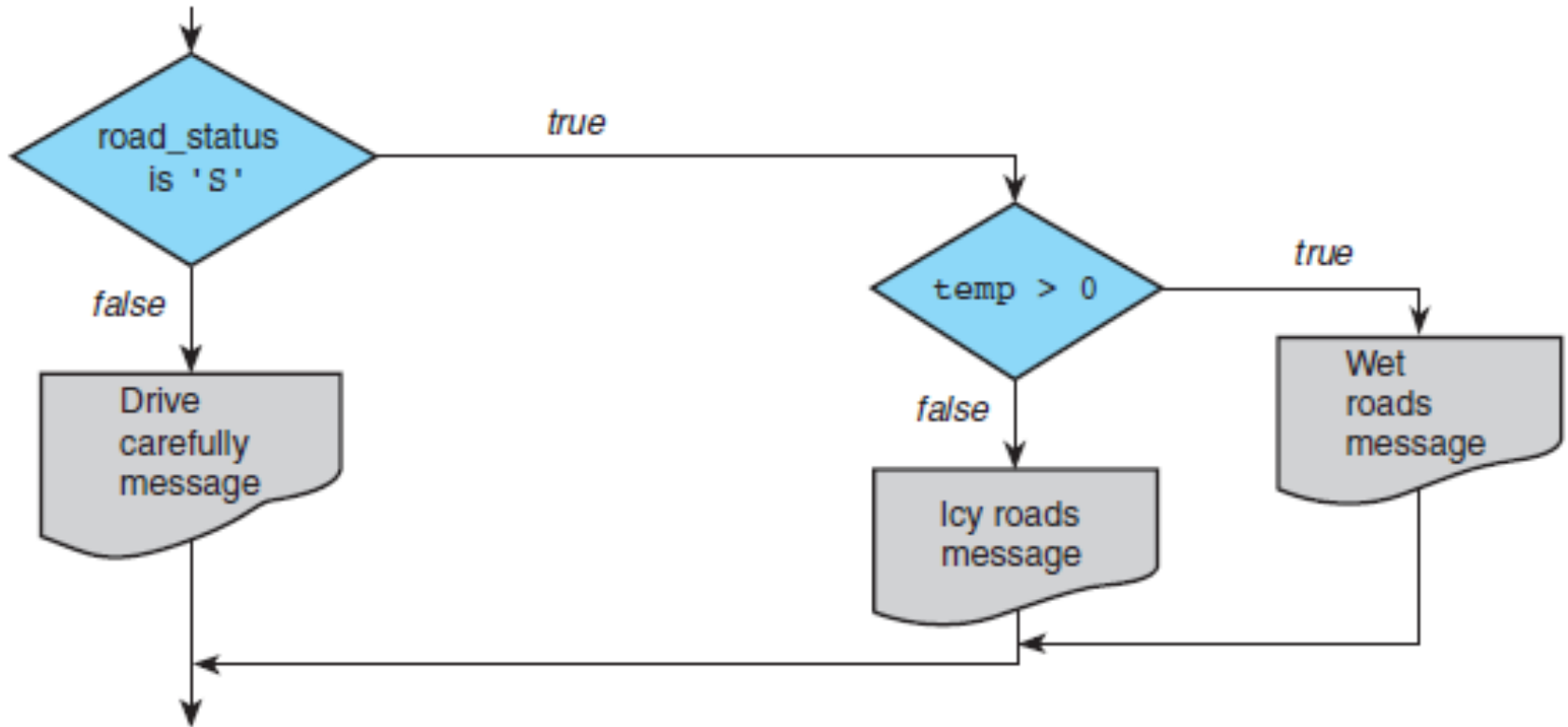
## Function comp\_tax

```
1. /*
2.  * Computes the tax due based on a tax table.
3.  * Pre : salary is defined.
4.  * Post: Returns the tax due for 0.0 <= salary <= 150,000.00;
5.  *       returns -1.0 if salary is outside the table range.
6.  */
7. double
8. comp_tax(double salary)
9. {
10.     double tax;
11.
12.     if (salary < 0.0)
13.         tax = -1.0;
14.     else if (salary < 15000.00)                /* first range      */
15.         tax = 0.15 * salary;
16.     else if (salary < 30000.00)                /* second range     */
17.         tax = (salary - 15000.00) * 0.18 + 2250.00;
18.     else if (salary < 50000.00)                /* third range      */
19.         tax = (salary - 30000.00) * 0.22 + 5400.00;
20.     else if (salary < 80000.00)                /* fourth range     */
21.         tax = (salary - 50000.00) * 0.27 + 11000.00;
22.     else if (salary <= 150000.00)              /* fifth range      */
23.         tax = (salary - 80000.00) * 0.33 + 21600.00;
24.     else
25.         tax = -1.0;
26.
27.     return (tax);
28. }
```

# Nested if-statements with more than one variable

```
if (road_status == 'S')
    if (temp > 0) {
        printf("Wet roads ahead\n");
        printf("Stopping time doubled\n");
    } else {
        printf("Icy roads ahead\n");
        printf("Stopping time quadrupled\n");
    }
else
    printf("Drive carefully!\n")
```

Figure 4.12  
Flowchart of Road Sign Decision Process





# 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

*controlling expression*



# Syntax

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

## Figure 4.13 Program Using a *switch* Statement for Selection

```
1. /*
2.  * Reads serial number and displays class of ship
3.  */
4.
5. #include <stdio.h>
6.
7. int
8. main(void)
9. {
10.     char class;      /* input - character indicating class of ship */
11.
12.     /* Read first character of serial number */
13.     printf("Enter ship serial number> ");
14.     scanf("%c", &class);      /* scan first letter */
15.
16.     /* Display first character followed by ship class */
17.     printf("Ship class is %c: ", class);
18.     switch (class) {
19.     case 'B':
20.     case 'b':
21.         printf("Battleship\n");
22.         break;
23.     case 'C':
24.     case 'c':
25.         printf("Cruiser\n");
26.         break;
27.     case 'D':
28.     case 'd':
29.         printf("Destroyer\n");
30.         break;
31.     case 'F':
32.     case 'f':
33.         printf("Frigate\n");
34.         break;
35.     default:
36.         printf("Unknown\n");
37.     }
38.
39.     return (0);
40. }
```

(continued)

# Figure 4.13

## Program Using a *switch* Statement for Selection (cont.)

---

### Sample Run 1

```
Enter ship serial number> f3456  
Ship class is f: Frigate
```

### Sample Run 2

```
Enter ship serial number> P210  
Ship class is P: Unknown
```

---

# 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.