Programming with C I

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.02.02

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.
- To learn how to write selection statements that choose between two alternatives in a program using the if statement.

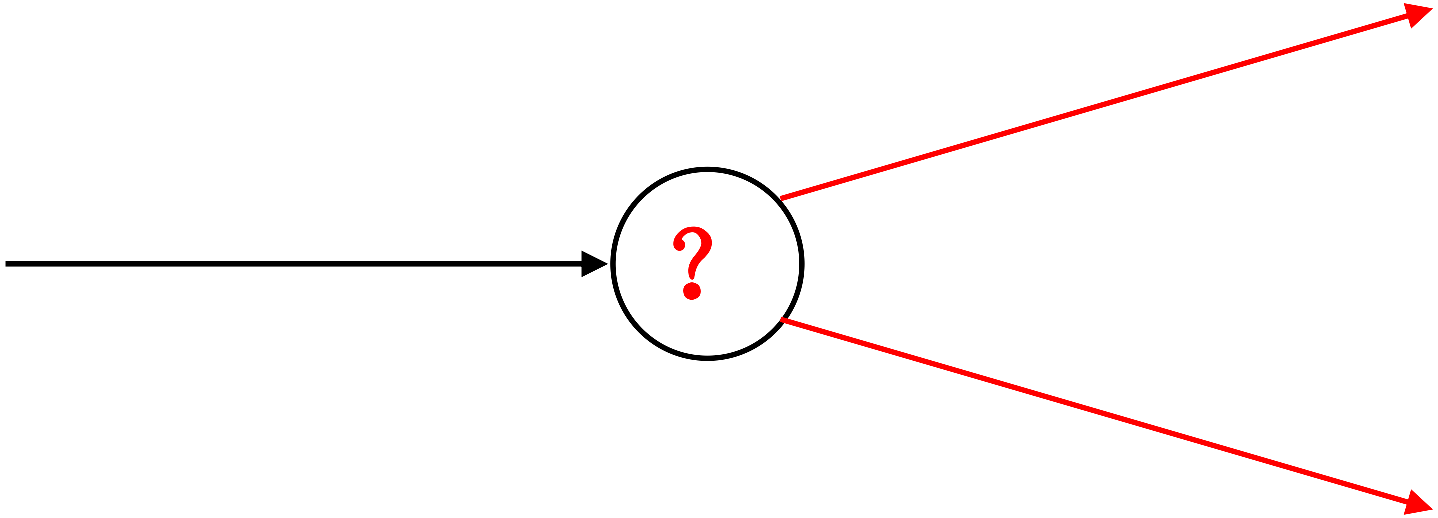
Objectives

- 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)

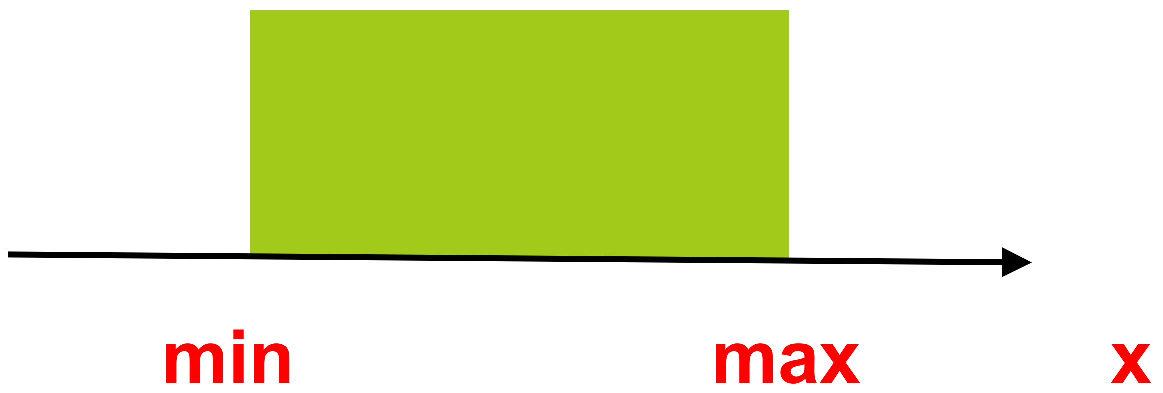
```
! (0 <= n && n <= 100)
```

Operator Precedence

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

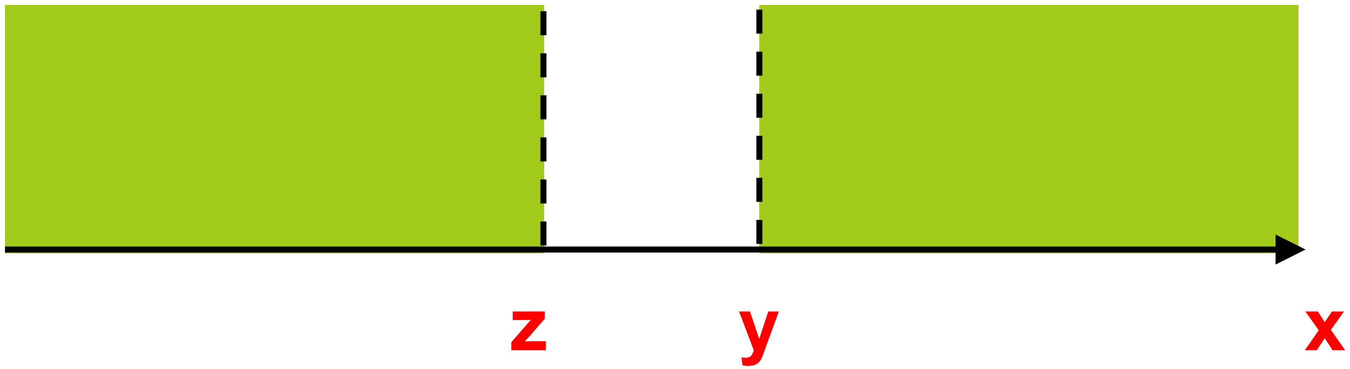
Figure

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

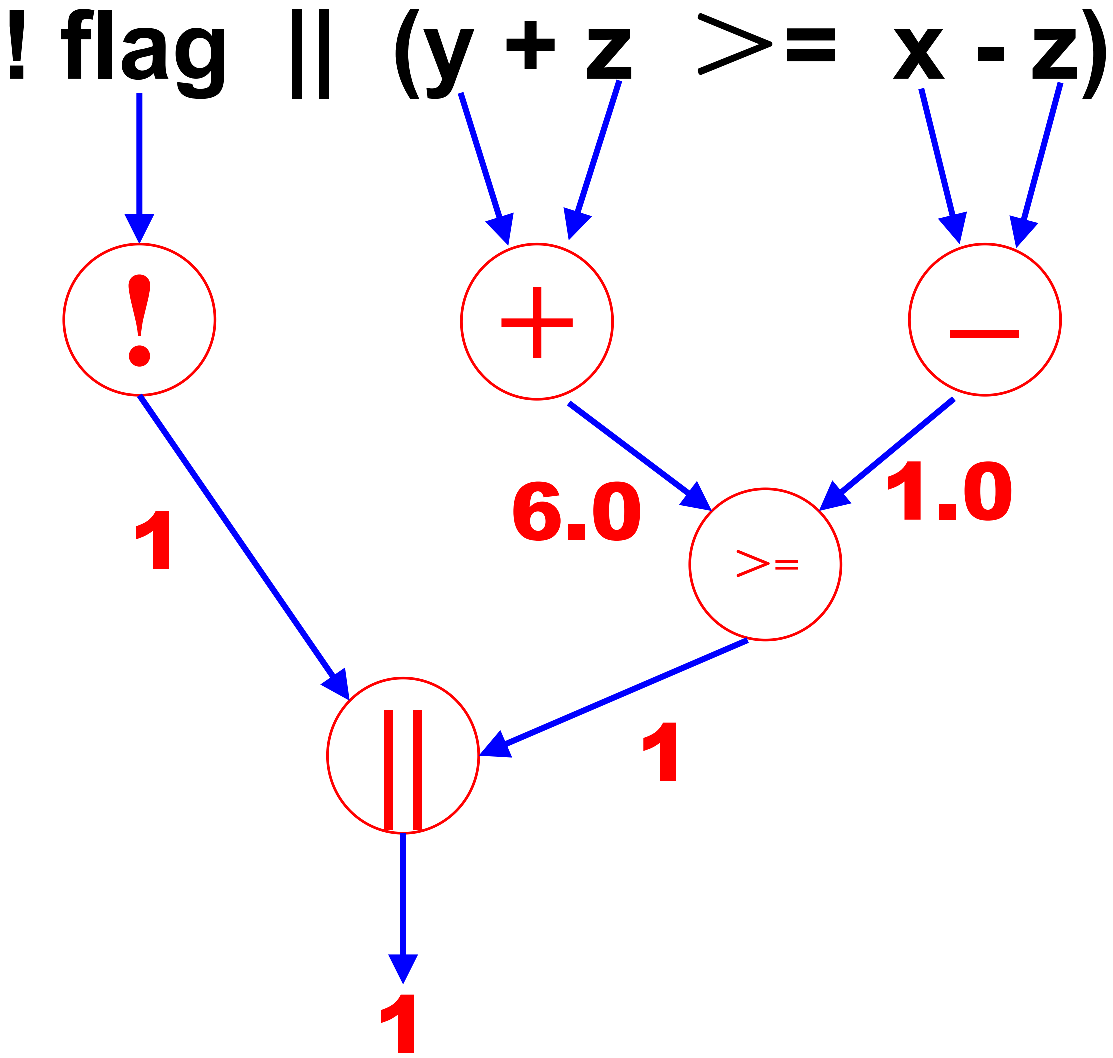


Figure

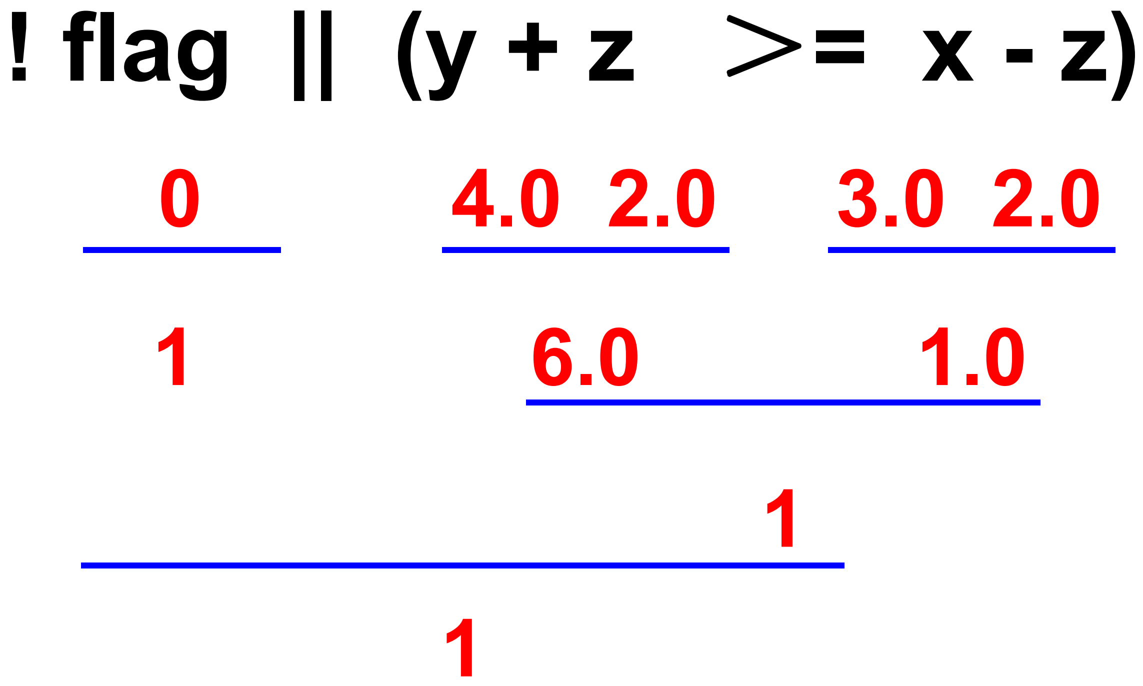
Range of True Values for z > x || x > y



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))
```

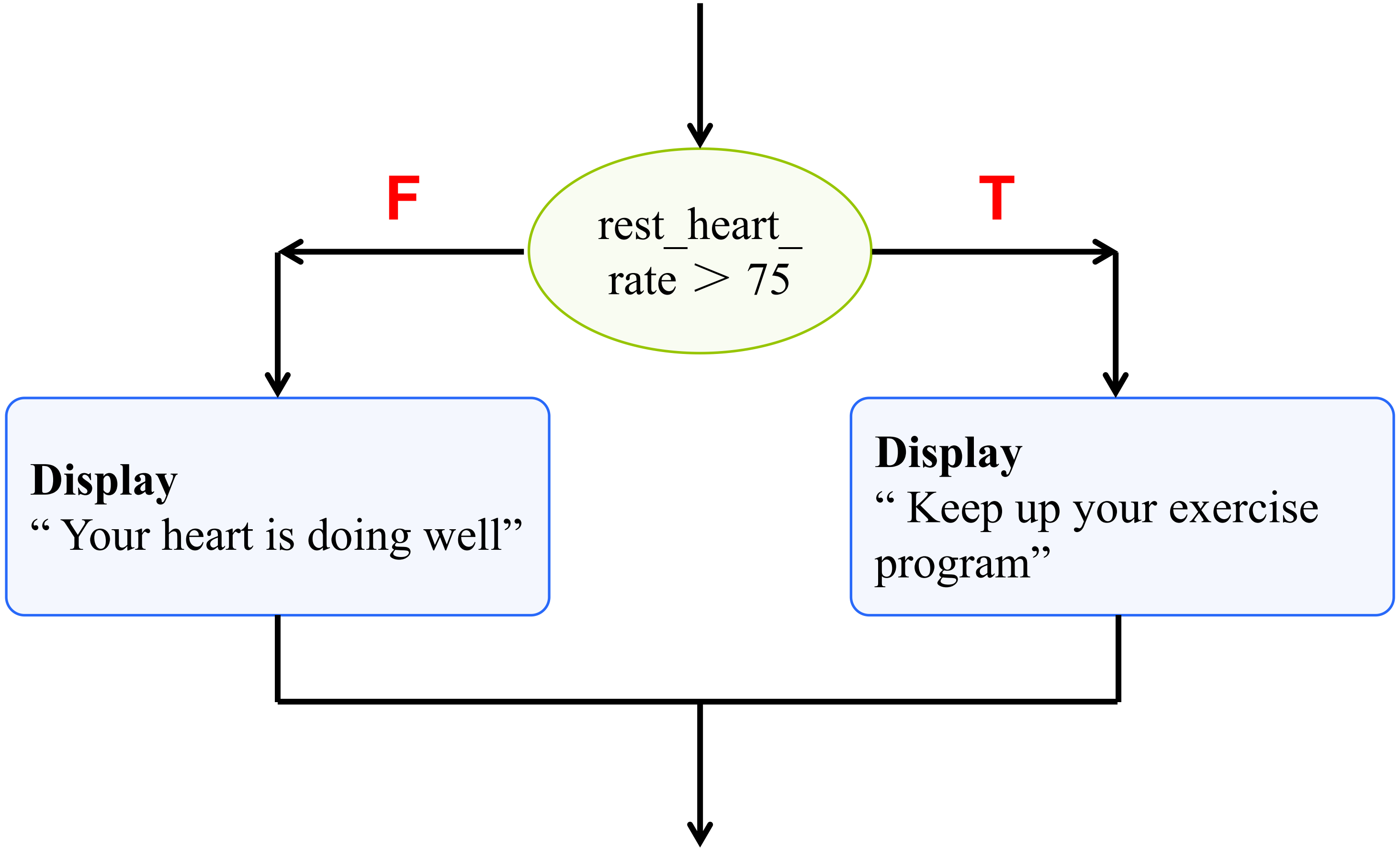
Comparing Characters

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

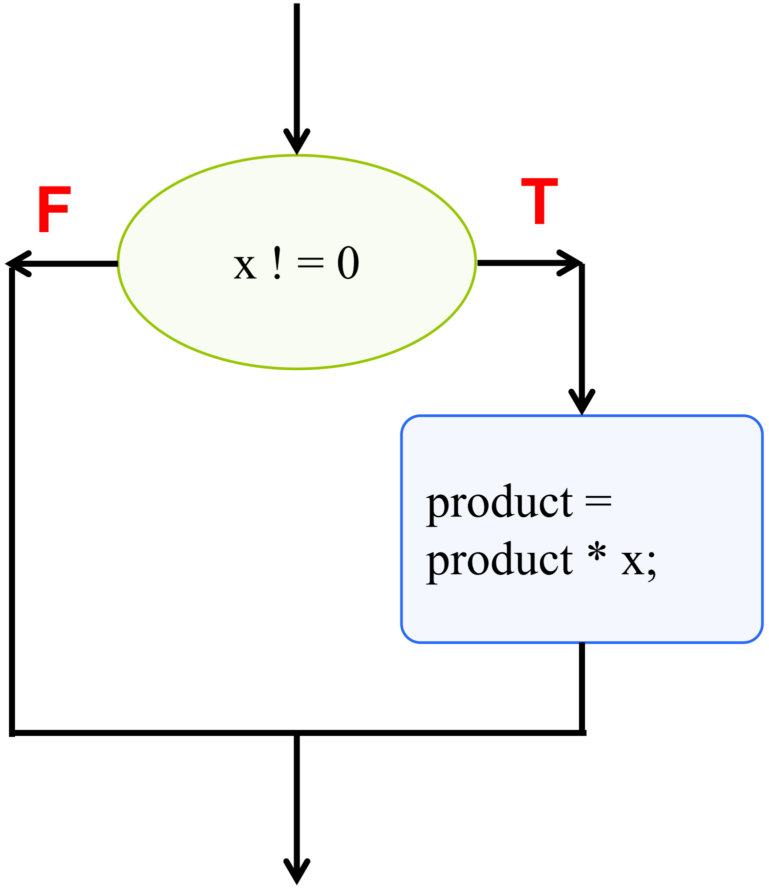
The if-statement

making decisions

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



(a)



(b)

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 Program Using an if statement for selection

```
/*
 * Displays message about heart rate.
 */
#include <stdio.h>

int main(void)
{
          int pulse;                          /* resting pulse rate for 10 secs */
          int rest_heart_rate;         /*  resting heart rate for 1 minute */

          /* Enter your resting pulse rate */
          printf(“Take your resting pulse for 10 seconds. \n”);
          printf(“Enter your pulse rate and press return>”);
          scanf(“%d”, &pulse);

          /* Calculate resting heart rate for minute */
          rest_heart_rate = pulse * 6
          printf(“Your resting heart rate is %d.\n”, rest_heart_rate);
          /* Display message based on resting heart rate */
         if (rest_heart_rate > 56)
            printf(“Keep up your exercise program!\n”);
         else
            printf(“Your heart is in excellent health!\n”);

          return (0);
}
```

(continued)

Figure Program Using an if statement for selection

```
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 if Statement to Order x and y

```
if (x > y)  {                     /* Switch x and y */
        temp = x;                 /* Store old x in temp */
        x = y;                    /* Store old y in x */
        y = temp;                 /* Store old x in y */
}
```

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
```

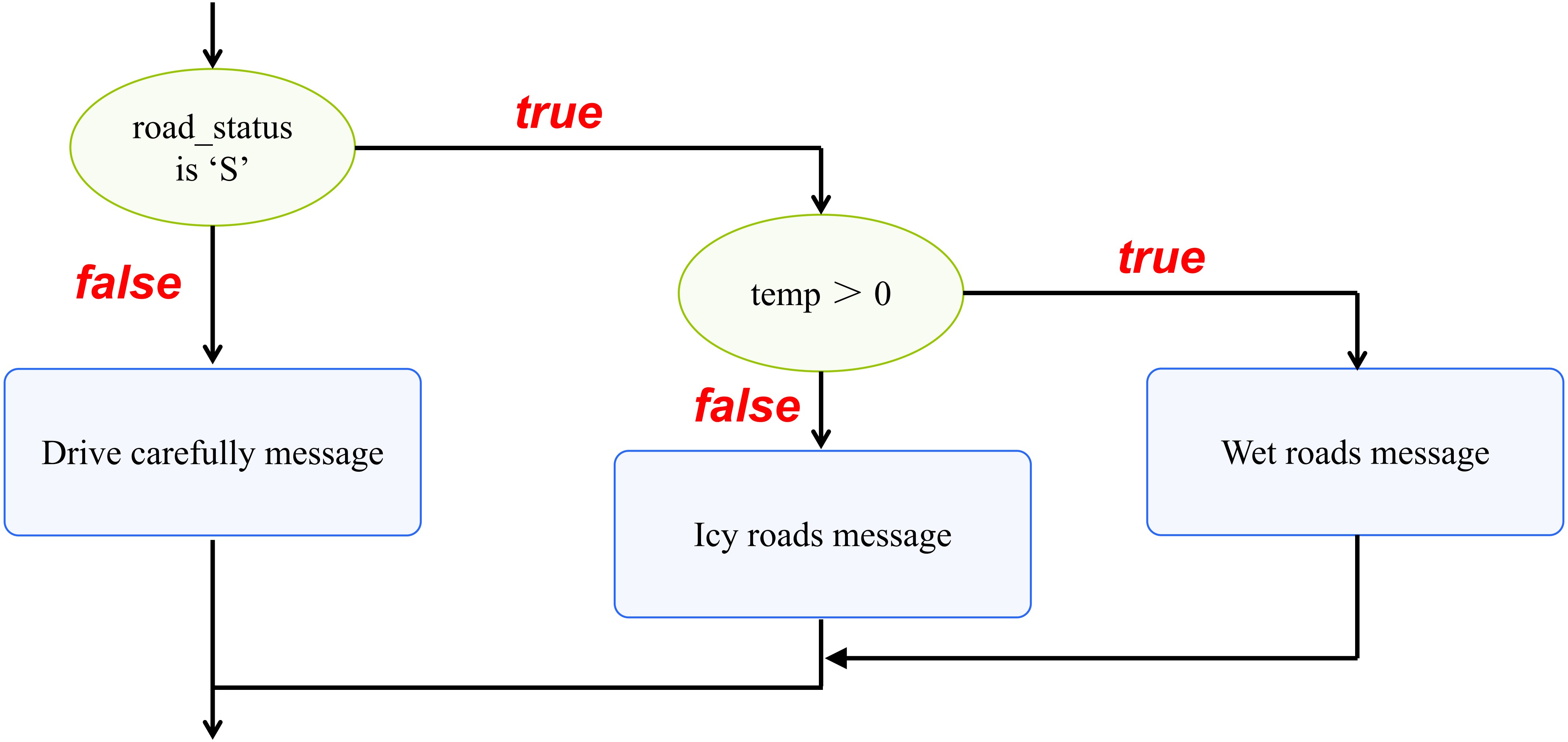
Figure Function comp_tax

```
/*
 * Computes the tax due based on a tax table.
 * Pre : salary is defined.
 * Post : Returns the tax due for 0.0 <= salary <= 150,000.00;
 *        returns -1.0 if salary is outside the table range.
 */
double
comp_tax(double salary)
{
      double tax;

      if (salary < 0.0)
            tax = -1.0;
      else if (salary < 15000.00)                    /* first range        */
            tax = 0.15 *salary;
      else if (salary < 30000.00)                    /* second range       */
             tax = (salary -15000.00) * 0.18 + 2250.00;
      else if (salary < 50000.00)                    /* third range        */
             tax = (salary - 30000.00) * 0.22 + 5400.00;
      else if (salary < 80000.00)                    /* fourth range       */
             tax = (salary - 50000.00) * 0.27 +11000.00;
       else if (salary <= 150000.00)                 /* fifth range        */
             tax = (salary -80000.00) * 0.33 + 21600.00;
        else
             tax = -1.0;

         return (tax)
}
```

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”)
```

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 ← **controlling expression**
- values may of type int or char
  - not double

Syntax

```
switch (controlling expression) {
            label set$_1$
                        statements$_1$
                        break;
            label set$_2$
                        statements$_2$
                        break;
                        .
                        .
                        .
            label set$_n$
                        statements$_n$
                        break;
```

Figure Program Using a *switch* Statement for Selection

```
/*
 * Reads serial number and displays class of ship
 */

#include <stdio.h>

int
main(void)
{
        char class;                                     /* input - character indicating class of ship */

        /* Read first character of serial number */
        printf(“Enter ship serial number>”);
        scanf(“%c”, &class);                            /* scan first letter */

        /* Display first character followed by ship class */
        printf(“Ship class is %c: ”, class);
        switch (class) {
        case ‘B’;
        case ‘b’;
                printf(“Battleship\n”);
                break;
```

(continued)

Figure Program Using a *switch* Statement for Selection

```
        case ‘C’;
        case ‘c’;
                printf(“Cruiser\n”);
                break;
        case ‘D’;
        case ‘d’;
                printf(“Destroyer\n”);
                break;
        case ‘F’;
        case ‘f’;
                printf(“Frigate\n”);
                break;
        default:
                printf(“Unknown\n”);
        }

        return (0);
}
```

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.

THE END

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.02.02