Lesson 6
Branch statements

Trịnh Thành Trung

trungtt@soict.hust.edu.vn

Topic of this week

- Branches
  - Class Lecture Review
    - If selection structure.
    - Switch selection structure.
  - Programming Exercises

The if Selection Structure

- Selection structure:
  - Used to choose among alternative courses of action
  - Pseudocode:

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

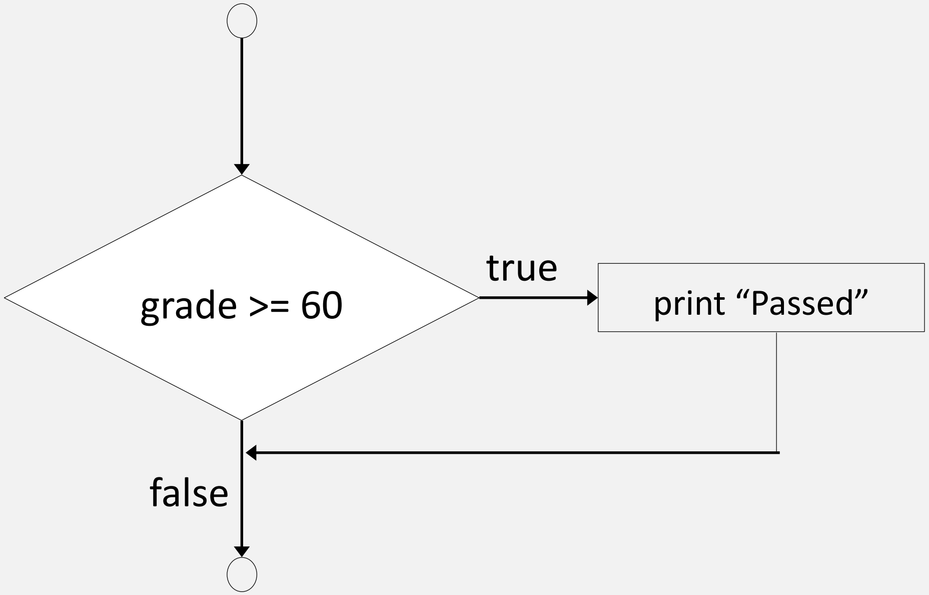
C code

```c
if ( grade >= 60 )
  printf( "Passed\n");
```

- If condition is true
  - Print statement executed and program goes on to next statement.
  - If false, print statement is ignored and the program goes onto the next statement.
  - Indenting makes programs easier to read
    + C ignores whitespace characters.

The if Selection Structure (II)

- Diamond symbol (decision symbol) - indicates decision is to be made
  - Contains an expression that can be true or false
  - Test the condition, follow appropriate path
- if structure is a single-entry/single-exit structure.



A decision can be made on any expression.

zero - **`false`**

nonzero - **`true`**

Example:

`3 - 4` is **`true`**

The if/else Selection Structure

- if
  - Only performs an action if the condition is true.
- if/else
  - A different action when condition is true than when condition is false
  - Psuedocode:

```
If student’s grade is greater than or equal to 60
  Print “Passed”
Else
  Print “Failed”
```

  - Note spacing/indentation conventions

C code

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

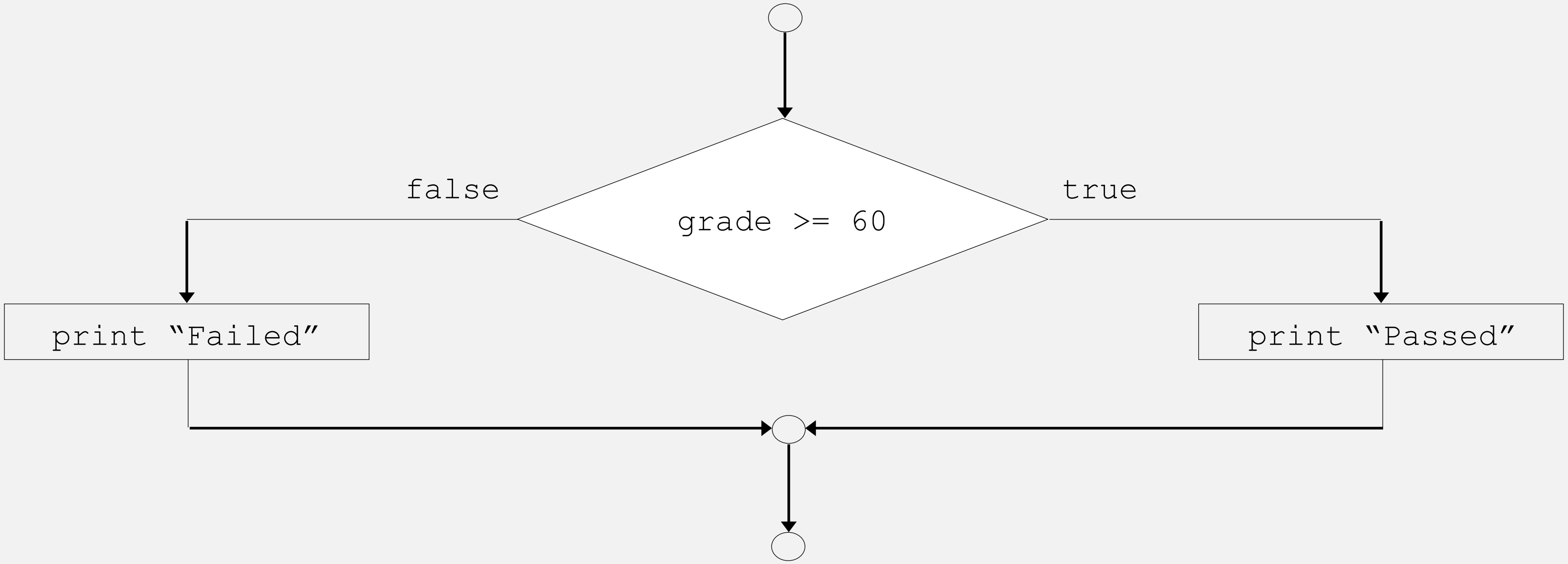
The if/else Selection Structure (II)

- Ternary conditional operator (**`?:`**)
  - Takes three arguments (condition, value if **`true`**, value if **`false`**)
  - Our pseudocode could be written:

```
printf( "%s\n", grade >= 60 ? "Passed" : "Failed" );
```

or

```
grade >= 60 ? printf( “Passed\n” ) : printf( “Failed\n” );
```



The if/else Selection Structure (III)

- Nested if/else structures
  - Test for multiple cases by placing if/else selection structures inside if/else selection structures

```
If student’s grade is greater than or equal to 90
Print “A”
else
If student’s grade is greater than or equal to 80
    Print “B”
else
    If student’s grade is greater than or equal to 70
        Print “C”
    else
        If student’s grade is greater than or equal to 60
            Print “D”
        else
            Print “F”
```

  - Once condition is met, rest of statements skipped
  - Deep indentation usually not used in practice

The if/else Selection Structure (IV)

- Compound statement:
  - Set of statements within a pair of braces
  - Example:

    ```
    if ( grade >= 60 )
       printf( "Passed.\n" );
    else {
       printf( "Failed.\n" );
       printf( "You must take this course again.\n" );
    }
    ```

  - Without the braces,

    ```
    printf( "You must take this course again.\n" );
    ```

    would be automatically executed
- Block: compound statements with declarations

The if/else Selection Structure (V)

- Syntax errors
  - Caught by compiler

- Logic errors:
  - Have their effect at execution time
  - Non-fatal: program runs, but has incorrect output
  - Fatal: program exits prematurely

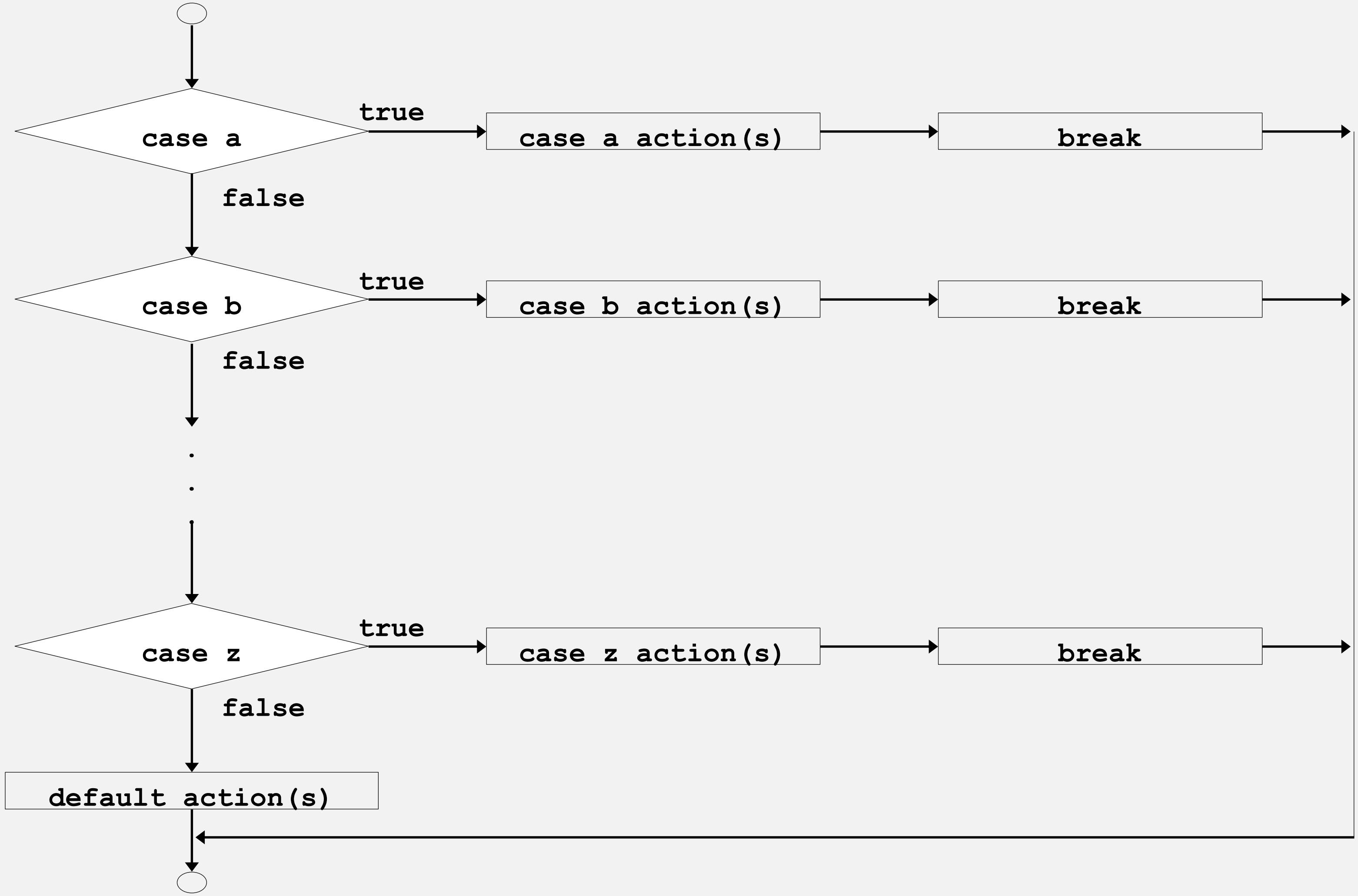
The switch Multiple-Selection Structure

- **`switch`**
  - Useful when a variable or expression is tested for all the values it can assume and different actions are taken.
- Format
  - Series of **`case`** labels and an optional **`default`** case

```
switch ( value ){
  case '1’:
    actions
  case '2’:
    actions
  default:
    actions
}
```

  - **`break;`** causes exit from structure

The switch Multiple-Selection Structure (II)



The switch Multiple-Selection Structure (III)

- Example of switch

```
c = getchar();
switch (c) {
    case '0': printf(“Zero\n”); break;
    case '1': case '2': case '3': case '4':
    case '5': case '6': case '7': case '8':
    case '9': printf(“Nine\n”);  break;
    case ' ':
    case '\n': newln++; break;
    case '\t': tabs++; break;
    default: printf(“missing char\n”); break;
}
```

Exercise 6.1

- Write a program that finds and displays the alphabetically first letter in a sequence (e.g. type IBK and it returns B).
- Note the use of a normal if/else and then an if on its own.

Exercise 6.2

- Write a program that transforms a compass heading to a compass bearing using this table:

HEADING IN DEGREES	BEARING COMPUTATION
0 - 89.999...	north (heading) east
90 - 179.999...	north (180.0 - heading) west
180 - 269.999...	south (heading - 180.0) west
270 - 360	South (360.0 - heading) east

Exercise 6.3

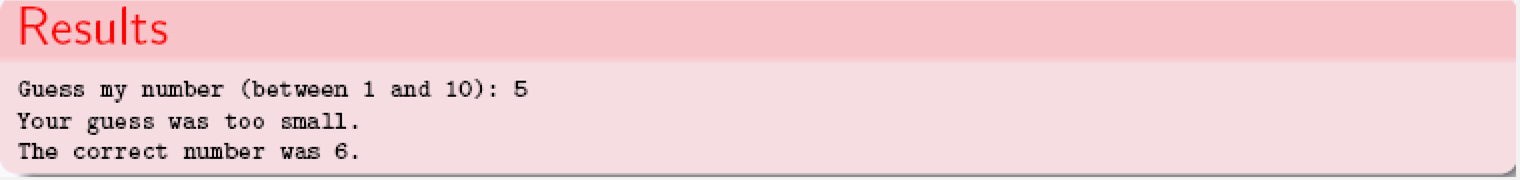
- Write a program that requires you enter an age and shows you what is your class. (Child, Senior Citizen or Adult)
  - Child : age <18
  - Adult : 18<=age<65
  - Senior Citizen: age >=65

Exercise 6.4

- Add a feature to program 6.3: buying cinema ticket. The policy is as follows:
  - Weekend: Basic price is $10 while other days basic price is $7.
  - Child: reduce 50%
  - Senior Citizen: reduce 30%
- Print the cinema ticket in this form
  - Movie: Avatar
  - Class: Child
  - Date: Weekend
  - Price: $5

Exercise 6.5

- Write a program to play “High/Low". The program “picks" a number. The human player tries to guess it. The program indicates if the guess is too high, too low, or correct. Then it stops.
- Sample outputs:



Results

```
Guess my number (between 1 and 10): 5
Your guess was too small.
The correct number was 6.
```

- Use rand() function to pick a random number.

How to generate a random number

```
#include <time.h>
#include <stdlib.h>

..
srand((unsigned)time(NULL));
rand() % M; /* generate 0 → M-1 */
```

Exercise 6.6

- Write a program that reads in three integers. and then determine which one is the smallest, and display it.
- If the values are a, b, and c, there are four cases:
  - a is smallest if a < b and a < c
  - b is smallest if b < a and b < c
  - c is smallest if c < a and c < b
  - No smallest When?

Exercise 6.7

- Alter the exercise 6.3 by using Switch selection structure.
- Alter the exercise 6.5 by using Switch selection structure.

Exercise 6.8

- Write a C program that does the following:
  - reads the type of a vehicle exiting a car park (C for car, B for bus and T for truck) and the number of hours spent in the car park.
  - calculates the parking fee given the following rates (GST included):
    - Car: $0.70/hr for the first 2 hours; $2.50/hr after 2 hours
    - Bus: $1.50/hr for the first 2 hours; $2.00/hr after 2 hours
    - Truck: $2.50/hr for the first hour; $3.25/hr after 1 hour
  - prints a request for payment that states the total parking fee owed.

```
PARKING FEE

VEHICLE – TYPE: CAR
TIME: 5.5
REGULAR FEE: 2*0.7 = 1.4
OvERTIME      : 3.5*2.5=7.5

TOTAL          :  8.9
Thank you.
```

Exercise 6.9. Simple Calculator

- Write a program that take the input from user that follows the syntax:
- operand operator operand
- operand: real value
- operator: + , -, *, /
- Return the result in the form:
  - a op b = c.
- For example:
  - 5+ 6 = 11;