

# IF Statements

Hui Chen

Department of Computer & Information Science


Brooklyn College

# Objectives

- To implement selection control using one-way **if** statements (§3.3).
- To implement selection control using two-way **if-else** statements (§3.4).
- To implement selection control using nested **if** and multi-way **if** statements (§3.5).
- To avoid common errors and pitfalls in **if** statements (§3.6).

# Outline

- Discussed
  - Boolean data type and Boolean expressions
- One-way if-statement
- Two-way if-statement
- Multi-way if-statement
- Nested if-statement
- Common errors and pitfalls



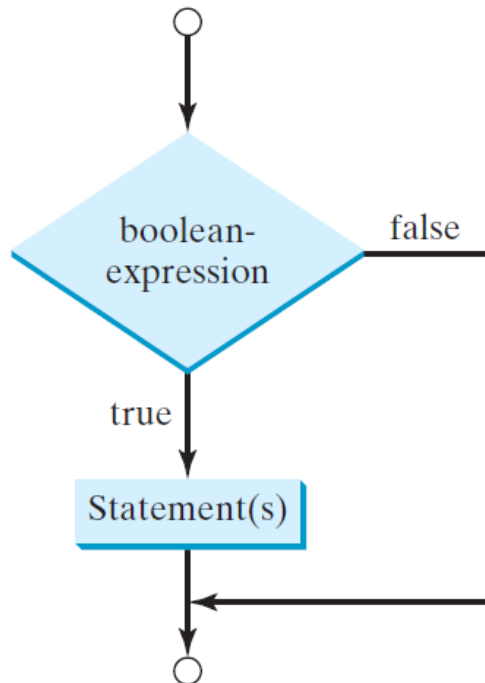
Flow Charts  
and  
Trace code

# Selecting Paths of Execution

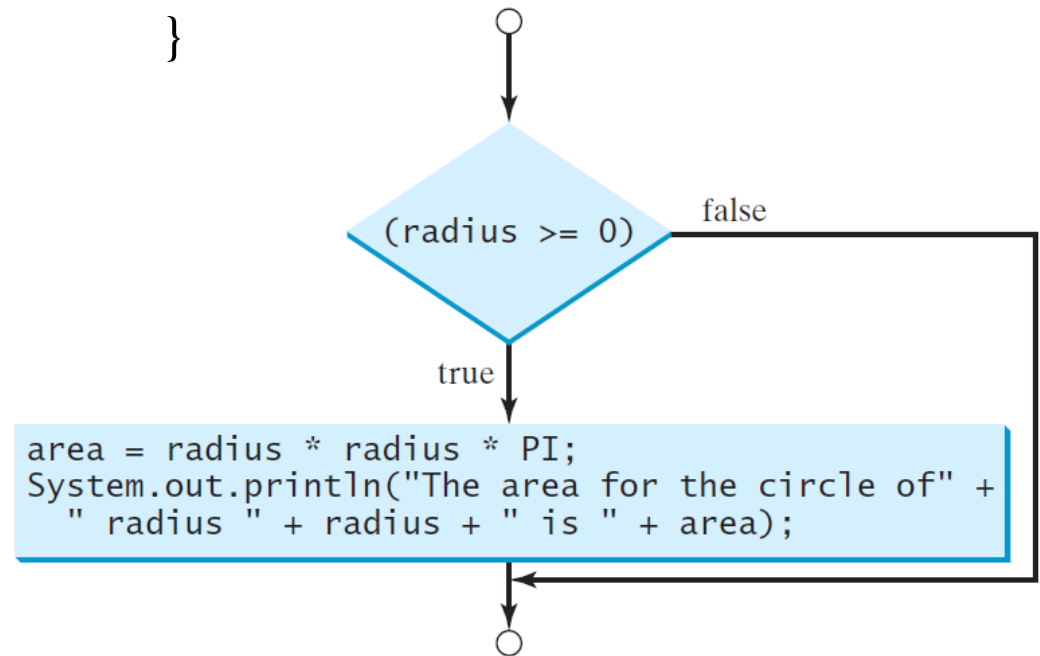
- The if statement is to specify alternative paths of decisions based on a Boolean expression
  - One-way if statements
  - Two-way if statements
  - Multi-way if statements
  - Nested if statements
- Let's begin with observing several one-way if-statement examples and draw flow charts ...

# One-way If-statement

```
if (boolean-expression) {  
    statement(s);  
}
```



```
if (radius >= 0) {  
    area = radius * radius * PI;  
    System.out.println("The area"  
        + " for the circle of radius "  
        + radius + " is " + area);  
}
```



# Let's observe ...

- Which path is being selected and executed?

# Which one is correct?

```
if i > 0 {  
    System.out.println("i is positive");  
}
```

(a)

```
if (i > 0) {  
    System.out.println("i is positive");  
}
```

(b)

# Which one is correct?

```
if (i > 0) {  
    System.out.println("i is positive");  
}
```

(a)

```
if (i > 0)  
    System.out.println("i is positive");
```

(b)



# To Have {} or not to have {}?

```
if (i > 0) {  
    System.out.println("i is positive");  
}
```

(a)

```
if (i > 0)  
    System.out.println("i is positive");
```

(b)

# More examples ...

# Example and Exercise 1

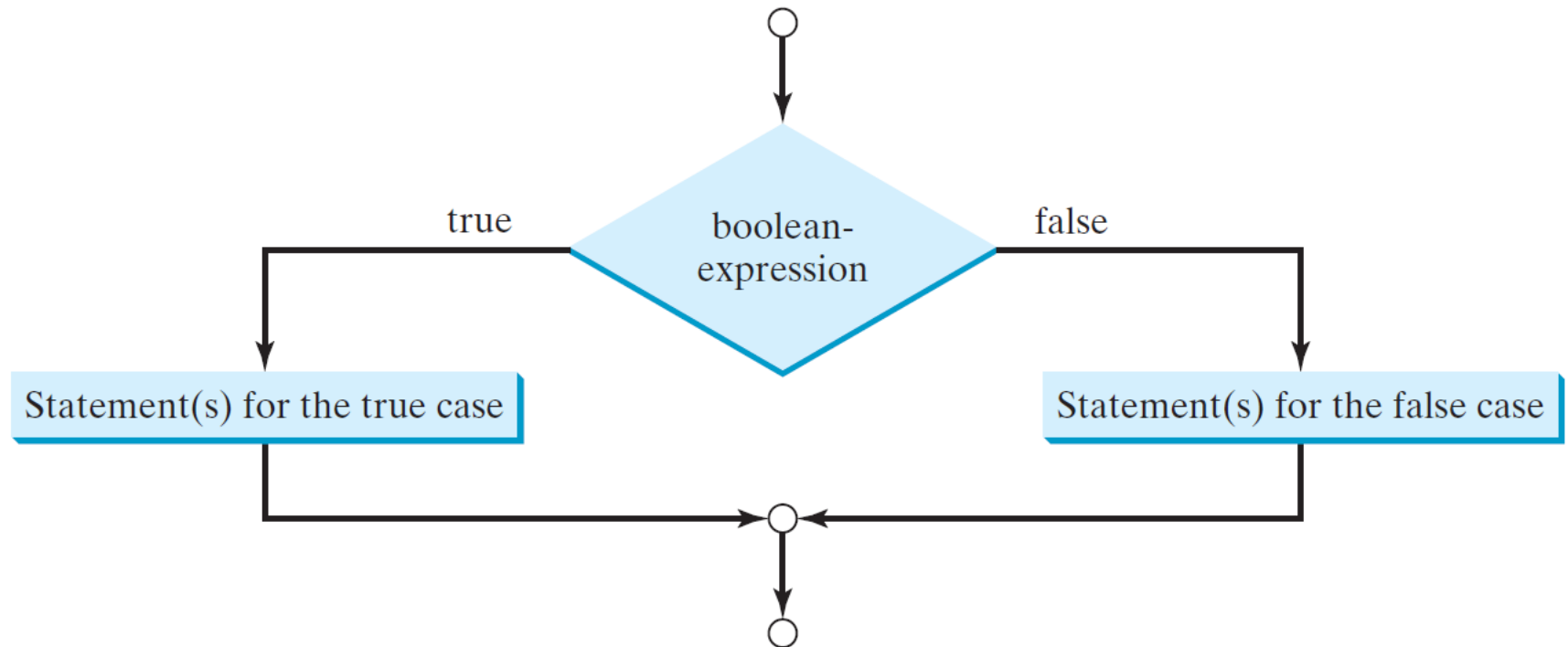
- Write a program that prompts the user to enter an integer. If the number is a multiple of 5, print HiFive. If the number is divisible by 2, print HiEven.

# Questions

# Two-way if statement (if-else)

```
if (boolean-expression) {  
    statement(s)-for-the-true-case;  
}  
else {  
    statement(s)-for-the-false-case;  
}
```

# Flow Chart of the Two-way if Statement



# Example & Exercise 2: Input

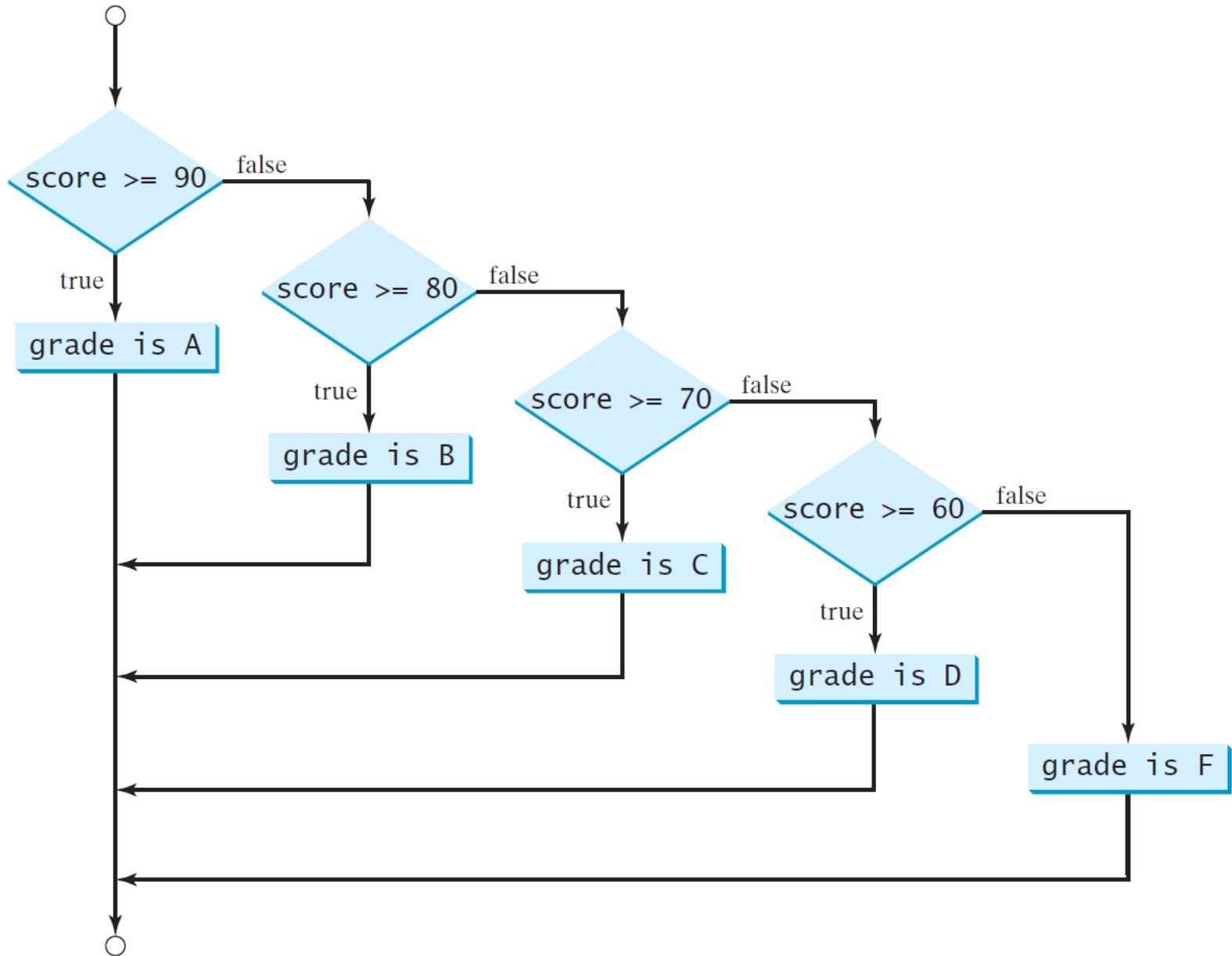
## Validation

```
if (radius >= 0) {  
    area = radius * radius * 3.14159;  
  
    System.out.println("The area for the "  
        + "circle of radius " + radius +  
        " is " + area);  
}  
else {  
    System.out.println("Negative input");  
}
```

# Multi-way If Statement

- Select among more than two execution paths
  - Multi-way if statement
  - Nested if statement
- How does it look like?





# Nested If Statements

```
if (score >= 90.0)
    System.out.print("A");
else
    if (score >= 80.0)
        System.out.print("B");
    else
        if (score >= 70.0)
            System.out.print("C");
        else
            if (score >= 60.0)
                System.out.print("D");
            else
                System.out.print("F");
```

# If-else if-else ...

```
if (score >= 90.0)
    System.out.print("A");
else if (score >= 80.0)
    System.out.print("B");
else if (score >= 70.0)
    System.out.print("C");
else if (score >= 60.0)
    System.out.print("D");
else
    System.out.print("F");
```

•

# Which one is better?

- They are equivalent, but ...

```
if (score >= 90.0)
    System.out.print("A");
else
    if (score >= 80.0)
        System.out.print("B");
    else
        if (score >= 70.0)
            System.out.print("C");
        else
            if (score >= 60.0)
                System.out.print("D");
            else
                System.out.print("F");
```

(a)

Equivalent

```
if (score >= 90.0)
    System.out.print("A");
else if (score >= 80.0)
    System.out.print("B");
else if (score >= 70.0)
    System.out.print("C");
else if (score >= 60.0)
    System.out.print("D");
else
    System.out.print("F");
```

(b)

# Example and Exercise 3

- Write a program that prompts the user to enter 3 integers and display the integers in non-decreasing (ascending) order

# Questions?

# Common Errors

- The else clause matches the most recent if clause in the same block.

```
int i = 1, j = 2, k = 3;
if (i > j)
  if (i > k)
    System.out.println("A");
else
  System.out.println("B");
```

(a)

Equivalent

This is better  
with correct  
indentation

```
int i = 1, j = 2, k = 3;
if (i > j)
  if (i > k)
    System.out.println("A");
  else
    System.out.println("B");
```

(b)

# Avoiding Common Errors

- Using indentations and {}



# Example: with or without {}

Nothing is printed from the preceding statement. To force the else clause to match the first if clause, you must add a pair of braces:

```
int i = 1;
int j = 2;
int k = 3;
if (i > j) {
    if (i > k)
        System.out.println("A");
}
else
    System.out.println("B");
```

This statement prints B.

# Common Errors

Adding a semicolon at the end of an if clause is a common mistake.

```
if (radius >= 0);  
{  
    area = radius*radius*PI;  
    System.out.println(  
        "The area for the circle of radius " +  
        radius + " is " + area);  
}
```

This mistake is hard to find, because it is not a compilation error or a runtime error, it is a logic error.

This error often occurs when you use the next-line block style.

# Questions?

# Tip

```
if (number % 2 == 0)
    even = true;
else
    even = false;
```

(a)

Equivalent

```
boolean even
= number % 2 == 0;
```

(b)

# Tip

```
if (even == true)
    System.out.println(
        "It is even.");
```

(a)

Equivalent

```
if (even)
    System.out.println(
        "It is even.");
```

(b)

# Questions?