Selection
Chapter 3 – Part 2

Course: CPSC 1150
Instructor: Dr. Bita Shadgar

Lecture 8

Learning Outcomes

- Implement java programs using if statement, if-else statement and nested if statements
- Apply switch statement to implement a specific group of nested if-else statements
- Debug and trace java programs including decision structures

Java syntax of decision structure

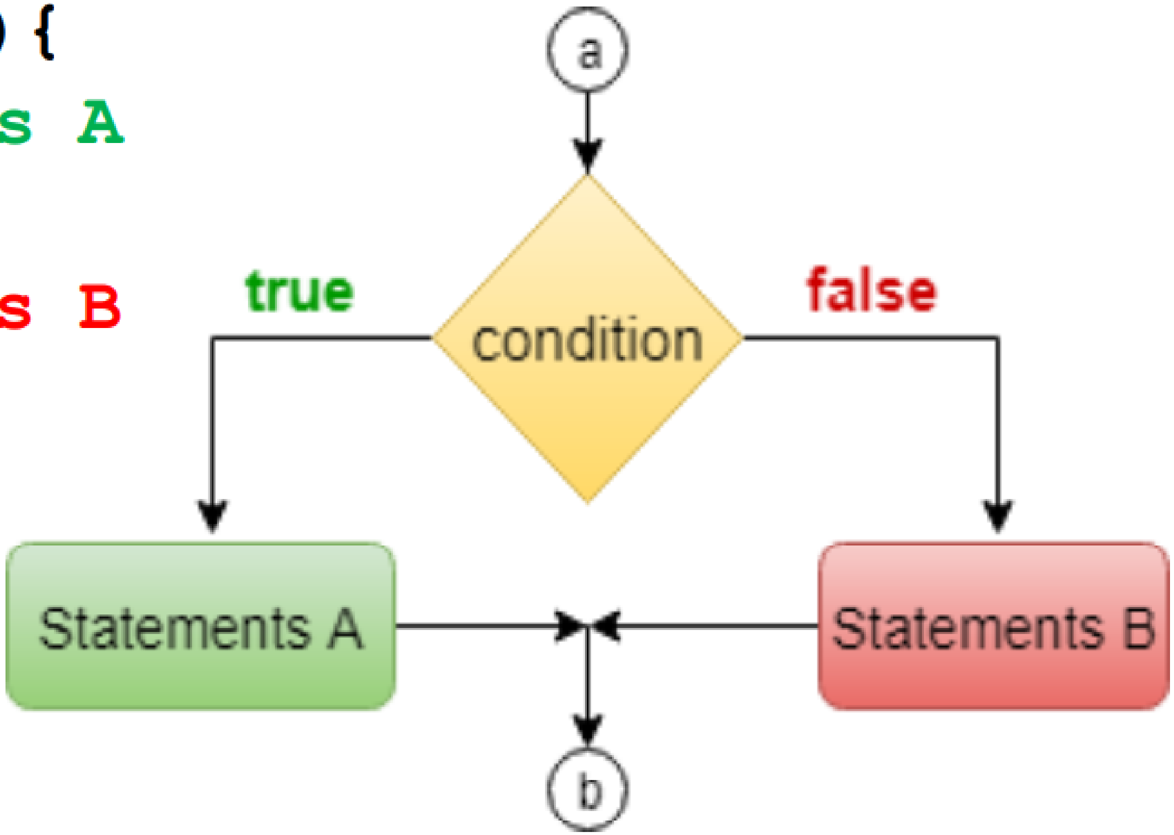
....

start from a

```
if (condition){
      Statements A
} else {
      Statements B
}
```

continue from b

...



Demo - Finding odd and even numbers

– Write a program that given a positive integer number less than 1000, prompts whether it is an odd or even number.

Question: How to modify the above code to work with a random number generated with computer?

Conditional operator

- Requires three expressions separated with a question mark and a colon
- Used as a short version of the if…else structure
- You are never required to use it, but it is good to concise the statement
- Syntax :

```
testExpression ? trueResult : falseResult;
```

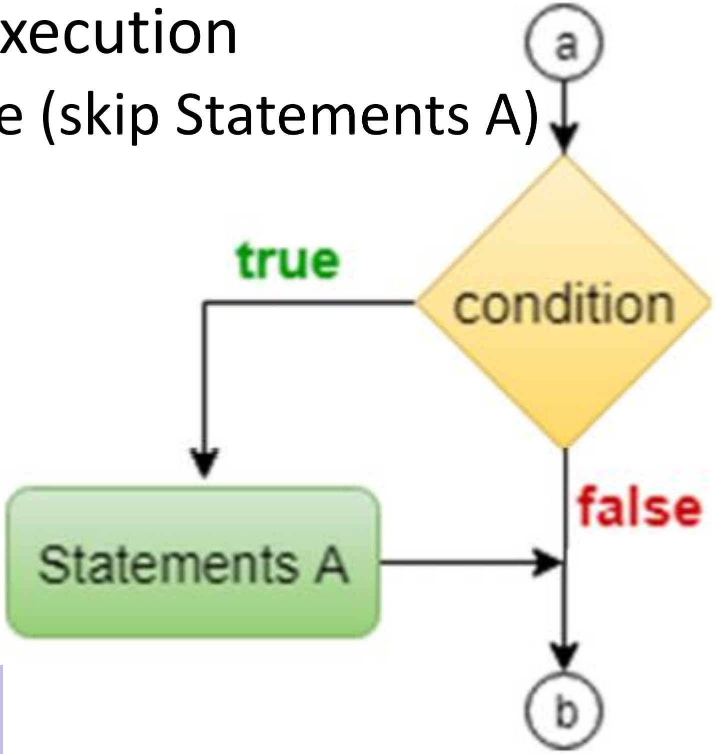
Example

```
String c = (x%2==0) ? x + " is even!" : x + " is odd";
System.out.println(c);
```

Single alternative decision structure

- Provides only one alternative path of execution
  - If condition is not true, exit the structure (skip Statements A)
- Java syntax:

```
…
if (condition)  {
      Statements A
}
…
```



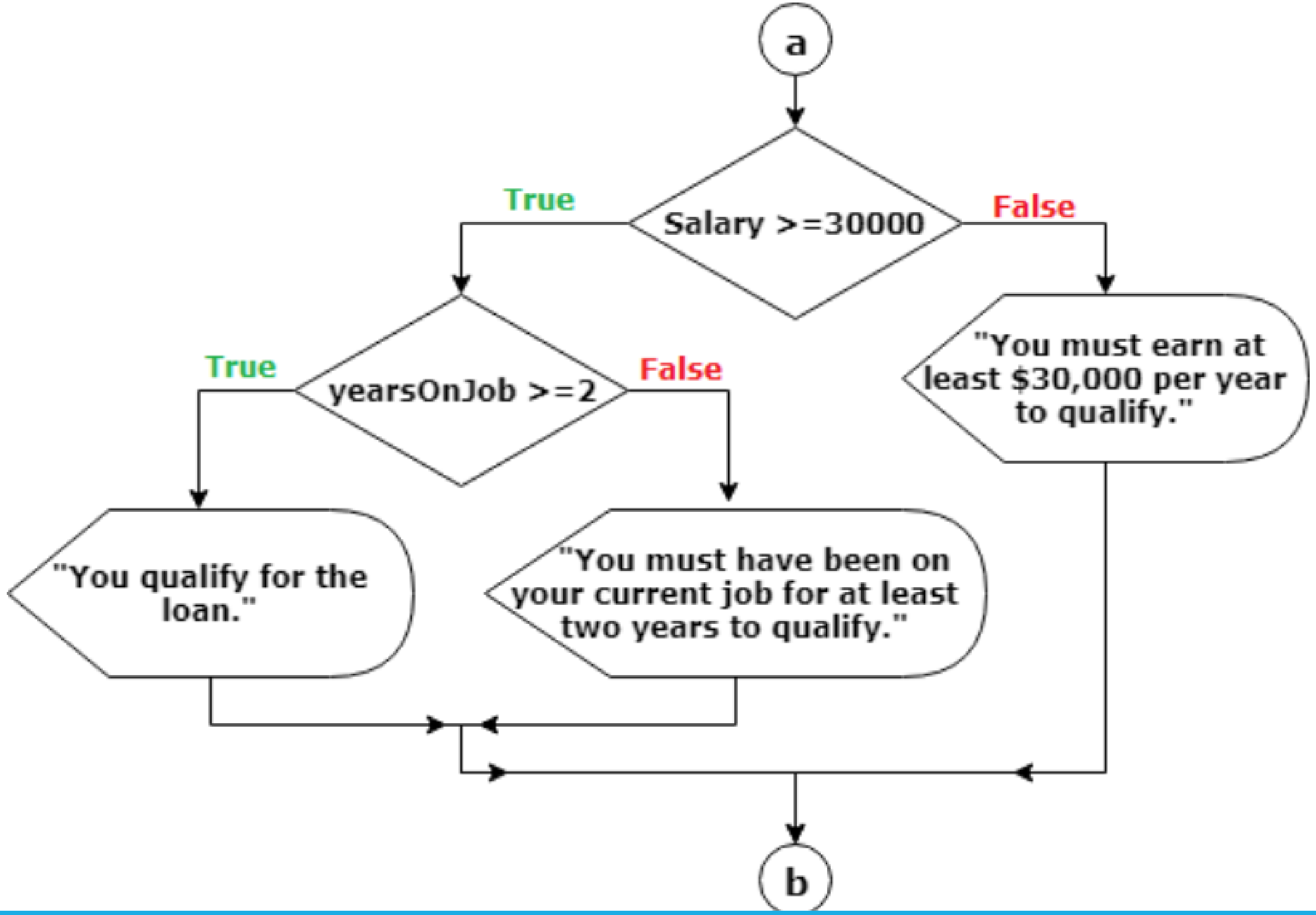
Example

```
/* Prints "Stay inside!" if the temperature is below zero */
if (temp < 0) {
      System.out.println("Stay inside! ");
}
```

Nested Decision Structures

- A decision structure can be nested inside another decision structure
  - Commonly needed in programs Example: Determine if someone qualifies for a loan, they must meet two conditions:
    1. Must earn at least $30,000/year
    2. Must have been employed for at least two years
  - Solution:
    - Check first condition, and if it is true, check second condition

Demo - Loan


a
Salary >=30000
True
False
yearsOnJob >=2
True
False
"You must earn at least $30,000 per year to qualify."
"You qualify for the loan."
"You must have been on your current job for at least two years to qualify."
b

If Else – rules and conventions

- If block has only one statement, curly braces are not mandatory
  - By convention, indentation should always be used to have a readable program (best practice)
- By default, an `else` matches with the most recent `if`
  - Curly braces can force different nesting

What happens? (OK Style)

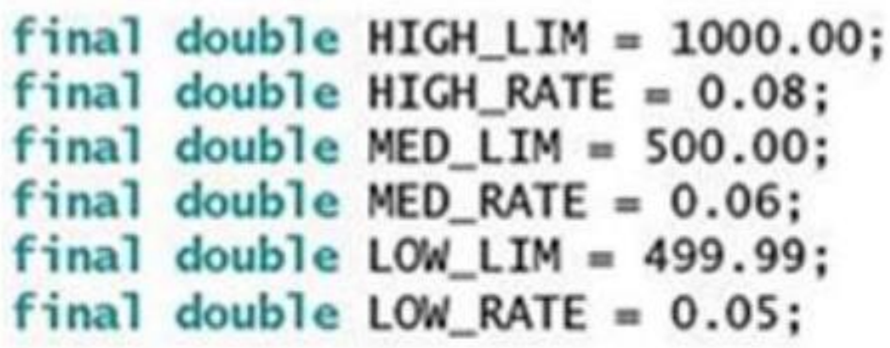
```
int i=4, j=1;
if (i<j)
   System.out.println("A");
else
   System.out.println("B");
```

How about here? (Bad style)

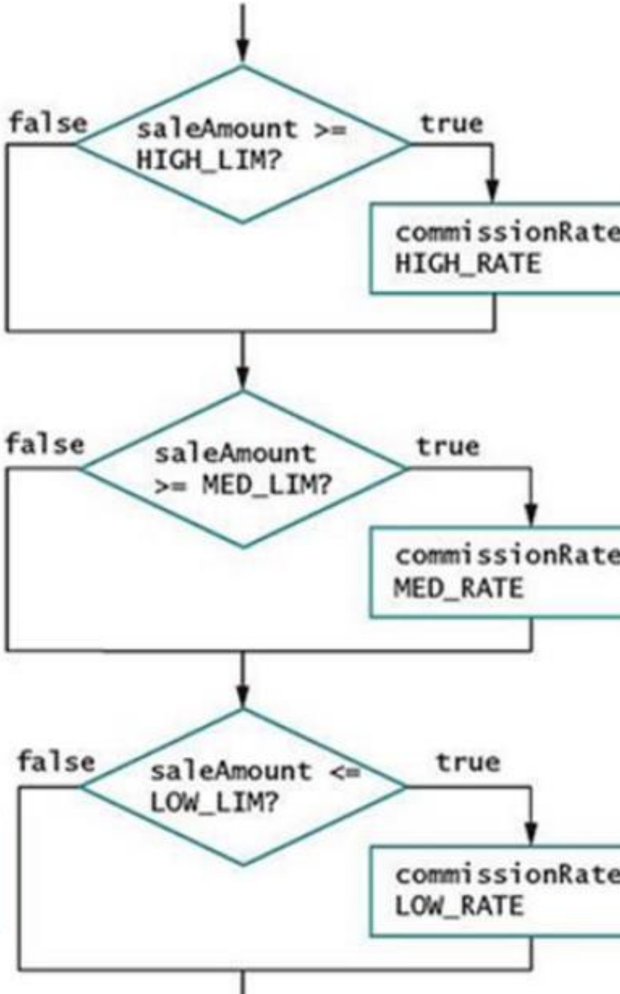
```
int i=4, j =1, k=3;
if (i < j)
   if (i < k)
        System.out.println ("A");
else
     System.out.println ("B");
```

Example – Sale’s commission rate

- Design an algorithm that decide the commission rate of sale people according to their sale amount. There is 8 percent commission If the sale amount is equal or more than $1000; 6 percent commission if sale amount is equal or more than $500; otherwise it is 5 percent commission.



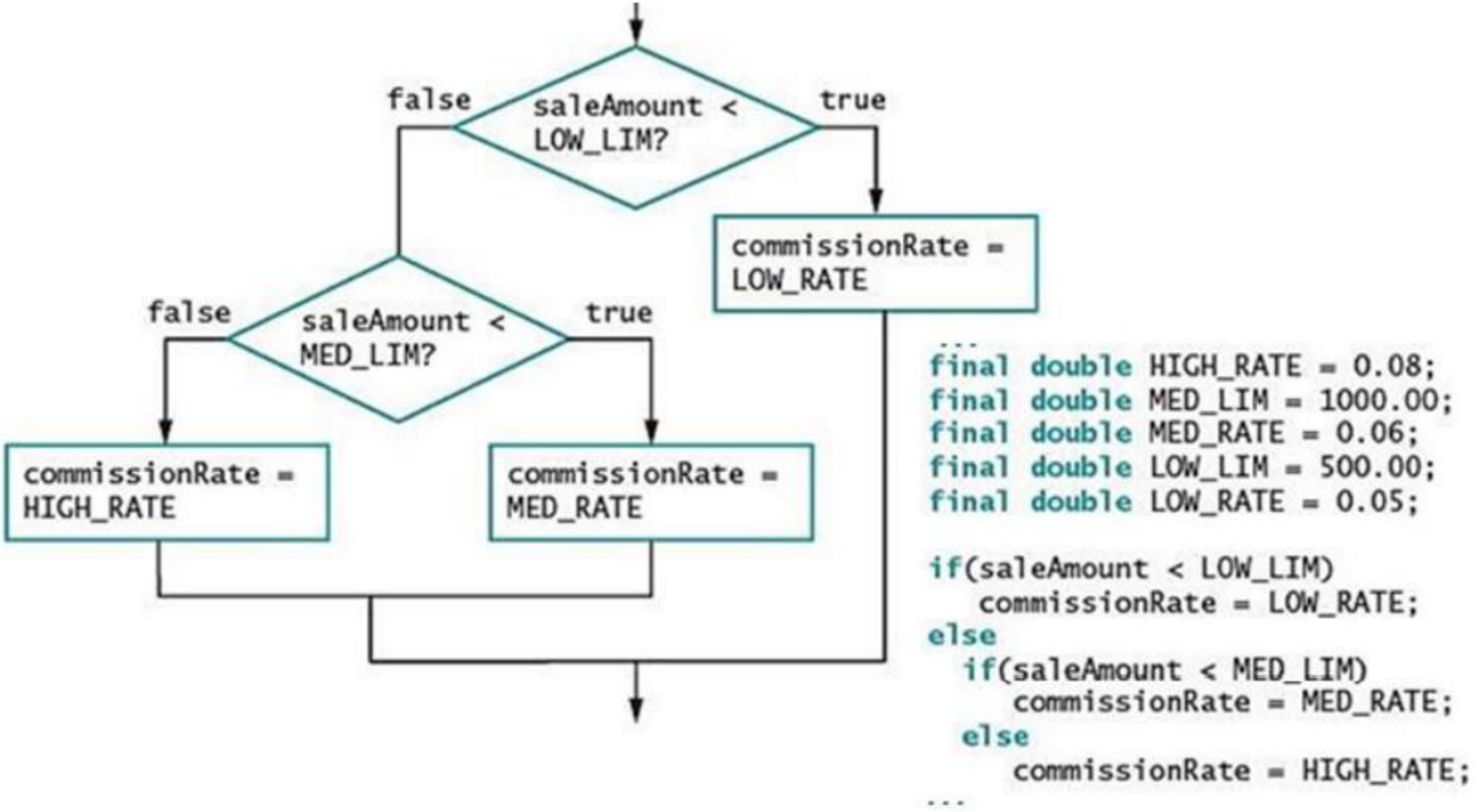
```
final double HIGH_LIM = 1000.00;
final double HIGH_RATE = 0.08;
final double MED_LIM = 500.00;
final double MED_RATE = 0.06;
final double LOW_LIM = 499.99;
final double LOW_RATE = 0.05;
```



Common mistake

Don’t do it : A high sale amount will result in medium sale commission, too.

Making Accurate and Efficient Decisions


false
saleAmount < LOW_LIM?
true
commissionRate = LOW_RATE
false
saleAmount < MED_LIM?
true
commissionRate = HIGH_RATE
commissionRate = MED_RATE

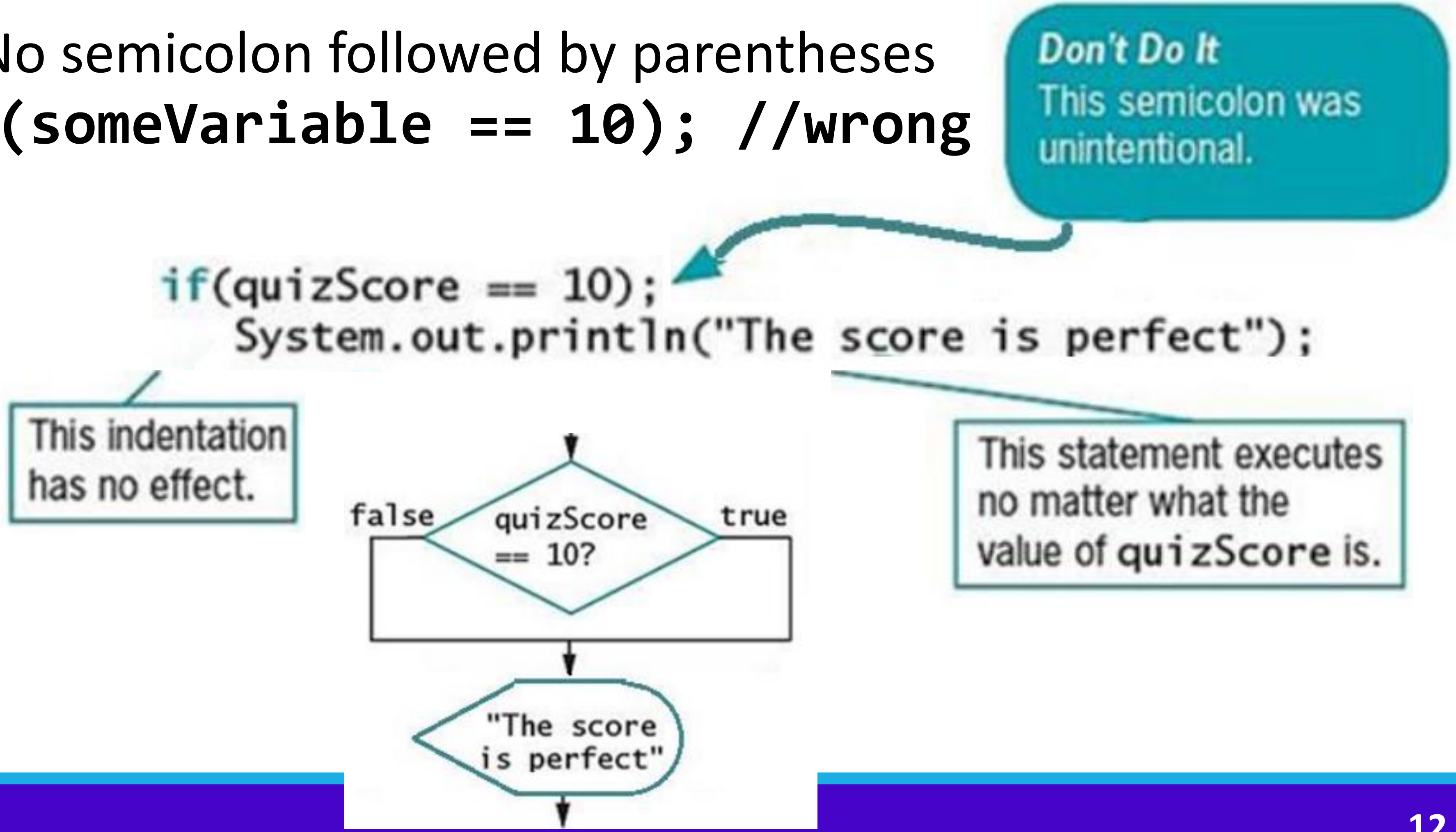
```
...
final double HIGH_RATE = 0.08;
final double MED_LIM = 1000.00;
final double MED_RATE = 0.06;
final double LOW_LIM = 500.00;
final double LOW_RATE = 0.05;

if(saleAmount < LOW_LIM)
   commissionRate = LOW_RATE;
else
   if(saleAmount < MED_LIM)
      commissionRate = MED_RATE;
   else
      commissionRate = HIGH_RATE;
...
```

Pitfall: Misplacing a semicolon

- An empty statement contains only a semicolon
- No semicolon followed by parentheses

```
if(someVariable == 10); //wrong
```



Don't Do It

This semicolon was unintentional.

```
if(quizScore == 10);
    System.out.println("The score is perfect");
```

This indentation has no effect.

This statement executes no matter what the value of quizScore is.

Pitfall: Assignment or equivalency operator

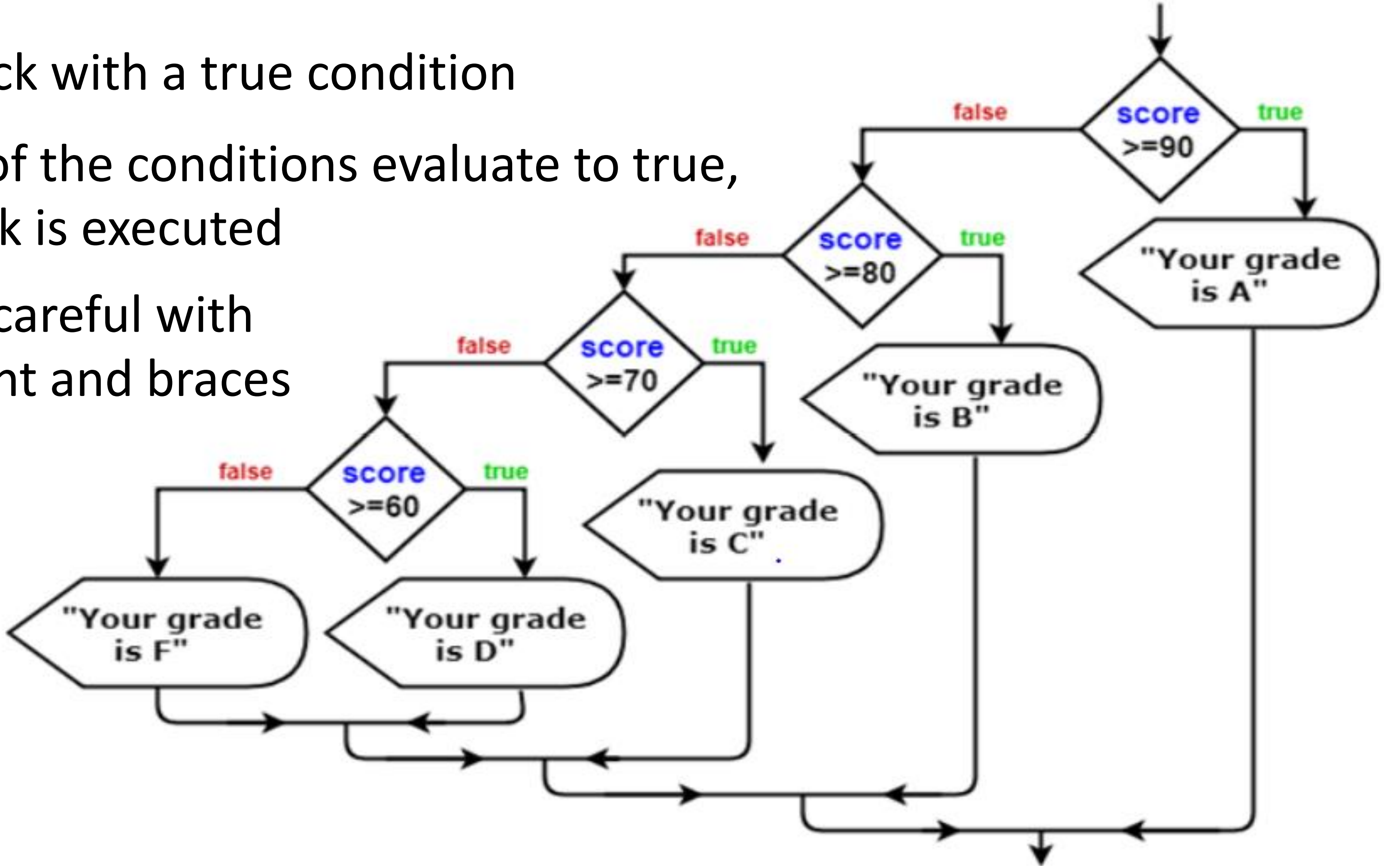
- Attempt to determine equivalency
  - Using a single equal sign rather than a double equal sign is illegal

- Check the output. Is this correct??

```
boolean x = false;
if (x = true) {// it should be x == true
     System.out.println("Inside if");
}
System.out.println("Outside if");
System.out.println(x);
```

Multi-way selections (nested if)

- Only one block is executed
- First block with a true condition
- If none of the conditions evaluate to true, last block is executed
- Be very careful with alignment and braces



Tracing

Definition

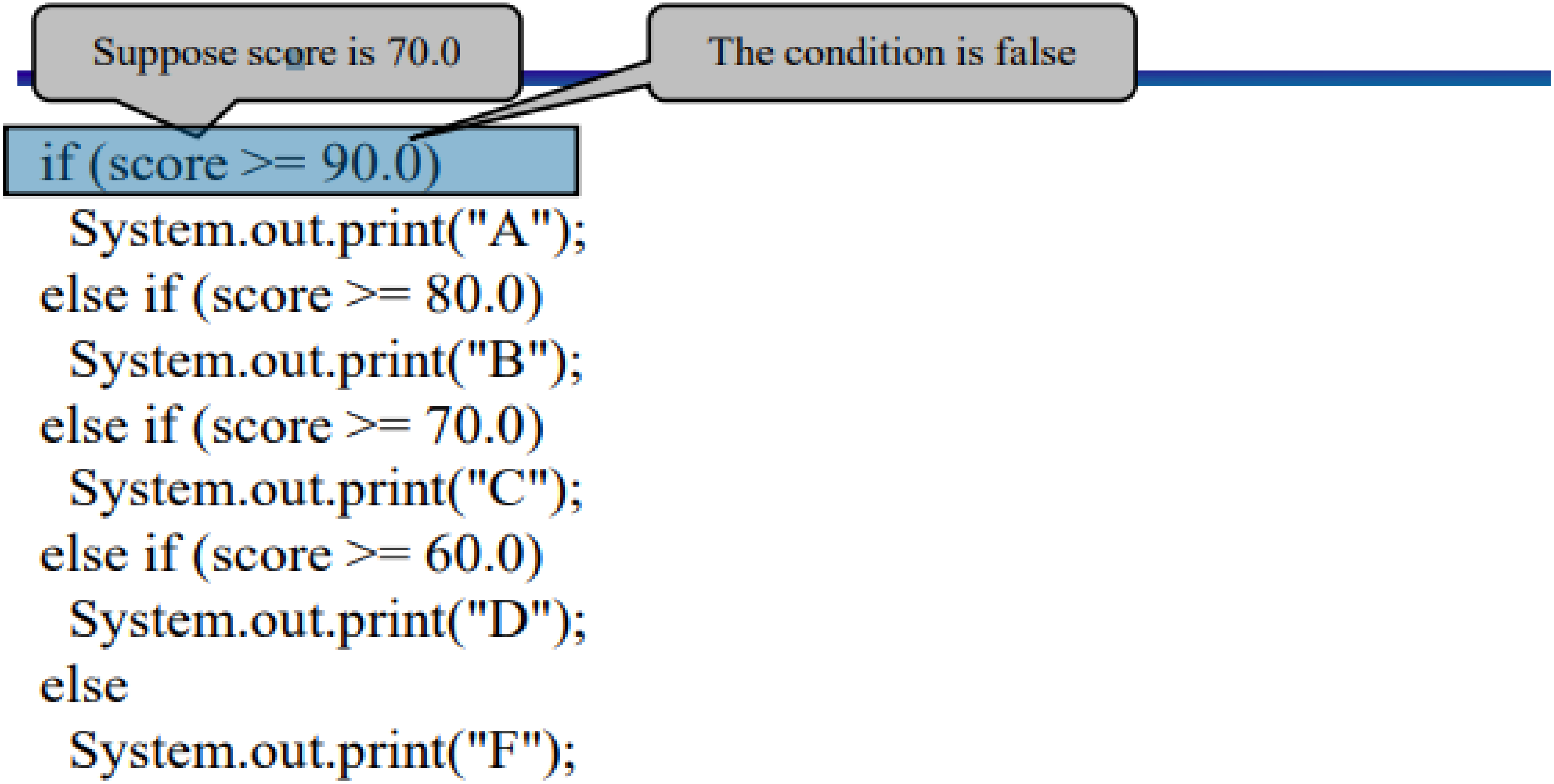
Tracing a program means predicting the flow of the program line by line, by looking at the source code.

- Tracing involves tracking values of variables as they change throughout the program

Example

Let’s write a program called CalcGrade with the same functionality as the flowchart on the previous slide. Before we run it, we will trace the program’s execution.

Trace if-else statement

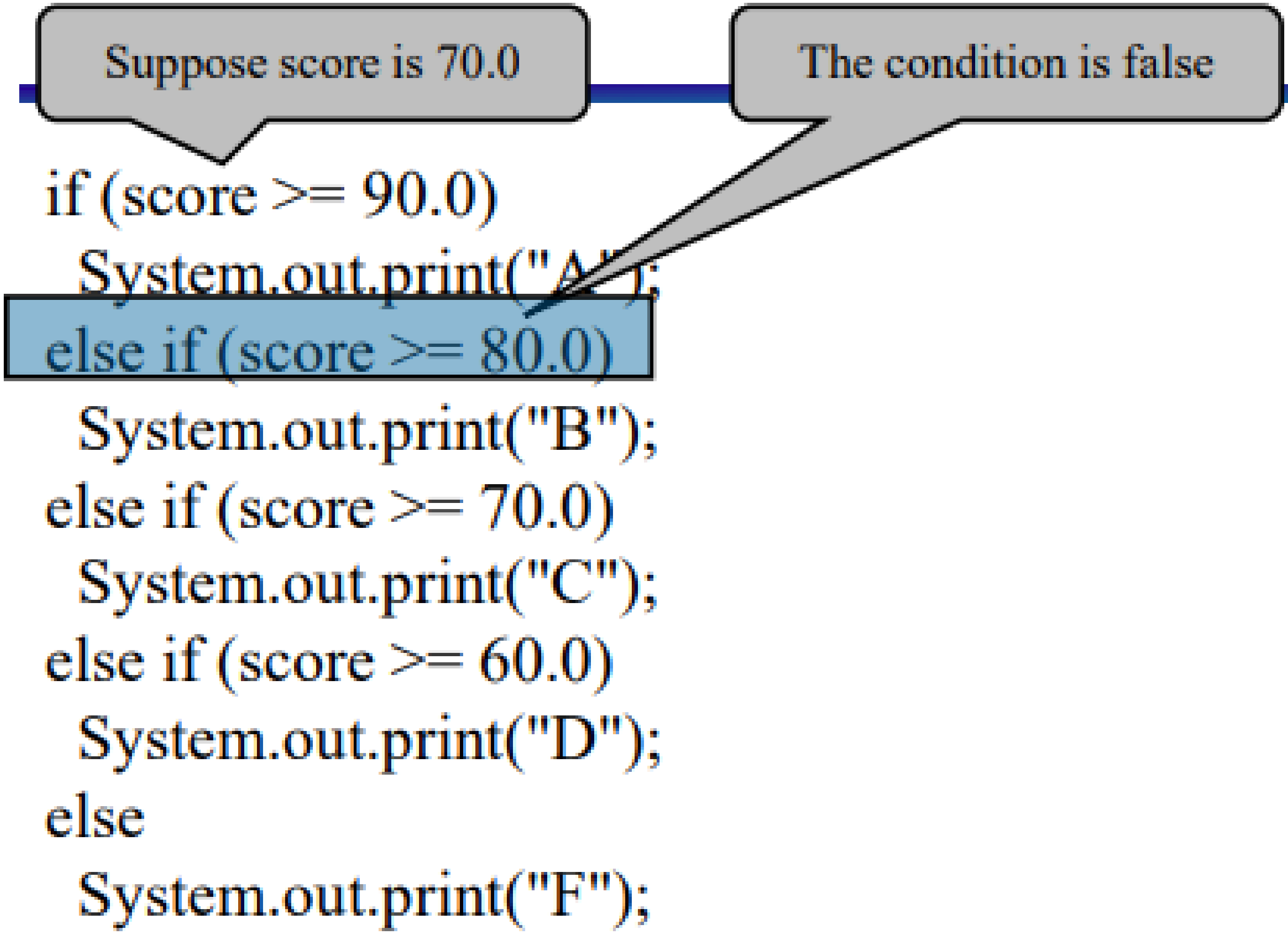


Suppose score is 70.0

The condition is false

```
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");
```

Trace if-else statement

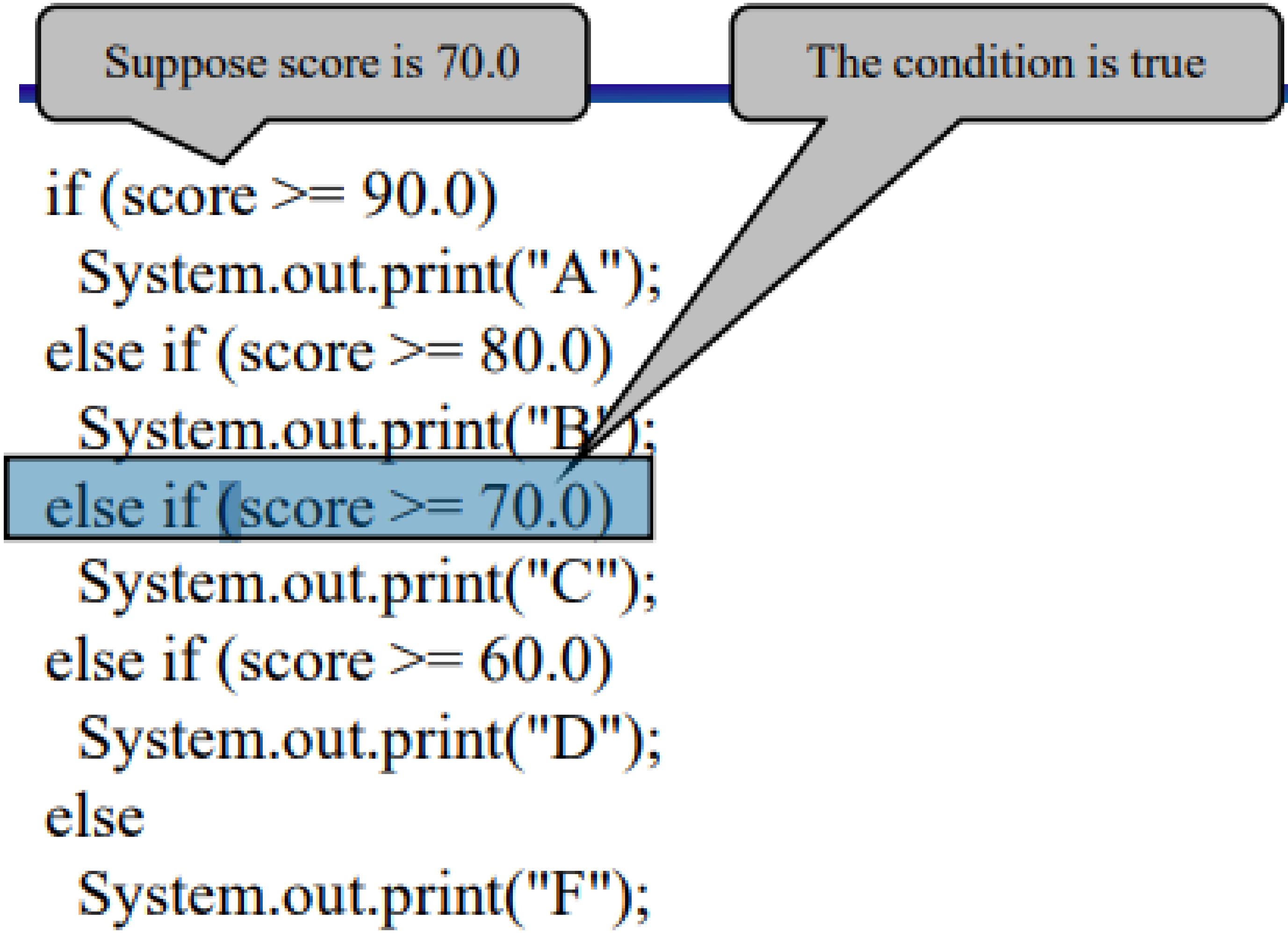


Suppose score is 70.0

The condition is false

```
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");
```

Trace if-else statement

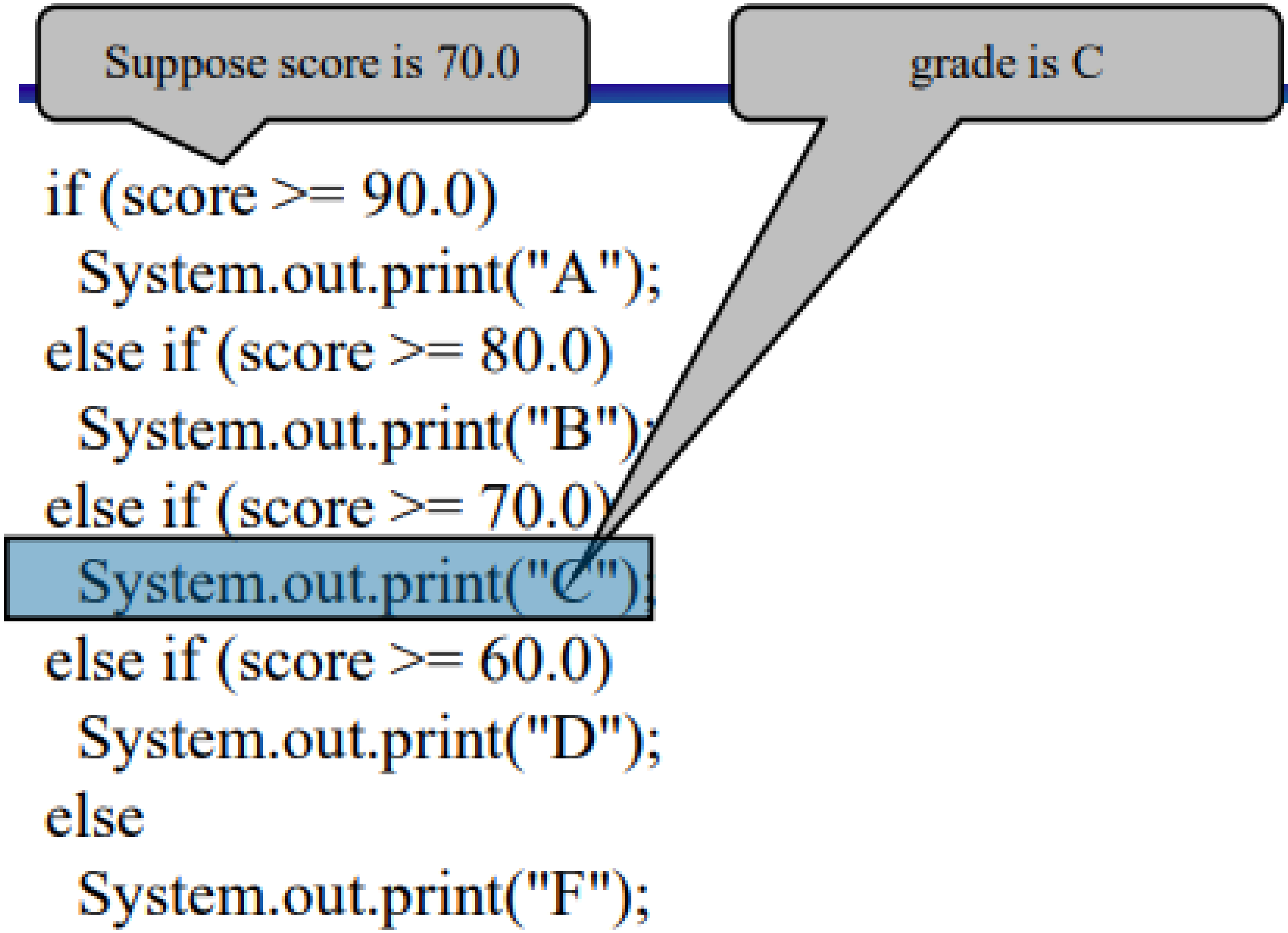


Suppose score is 70.0

The condition is true

```
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");
```

Trace if-else statement

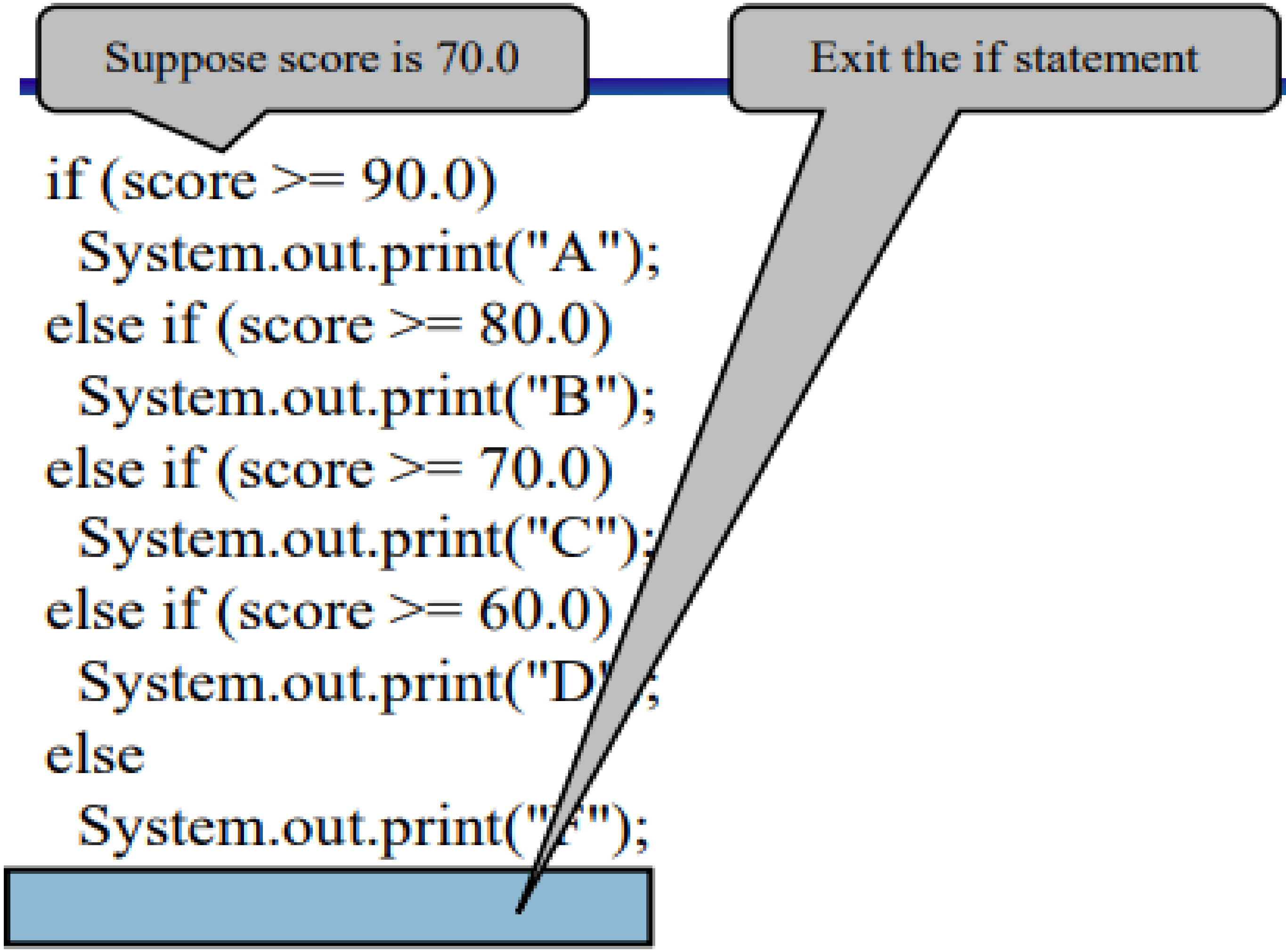


Suppose score is 70.0

grade is C

```
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");
```

Trace if-else statement



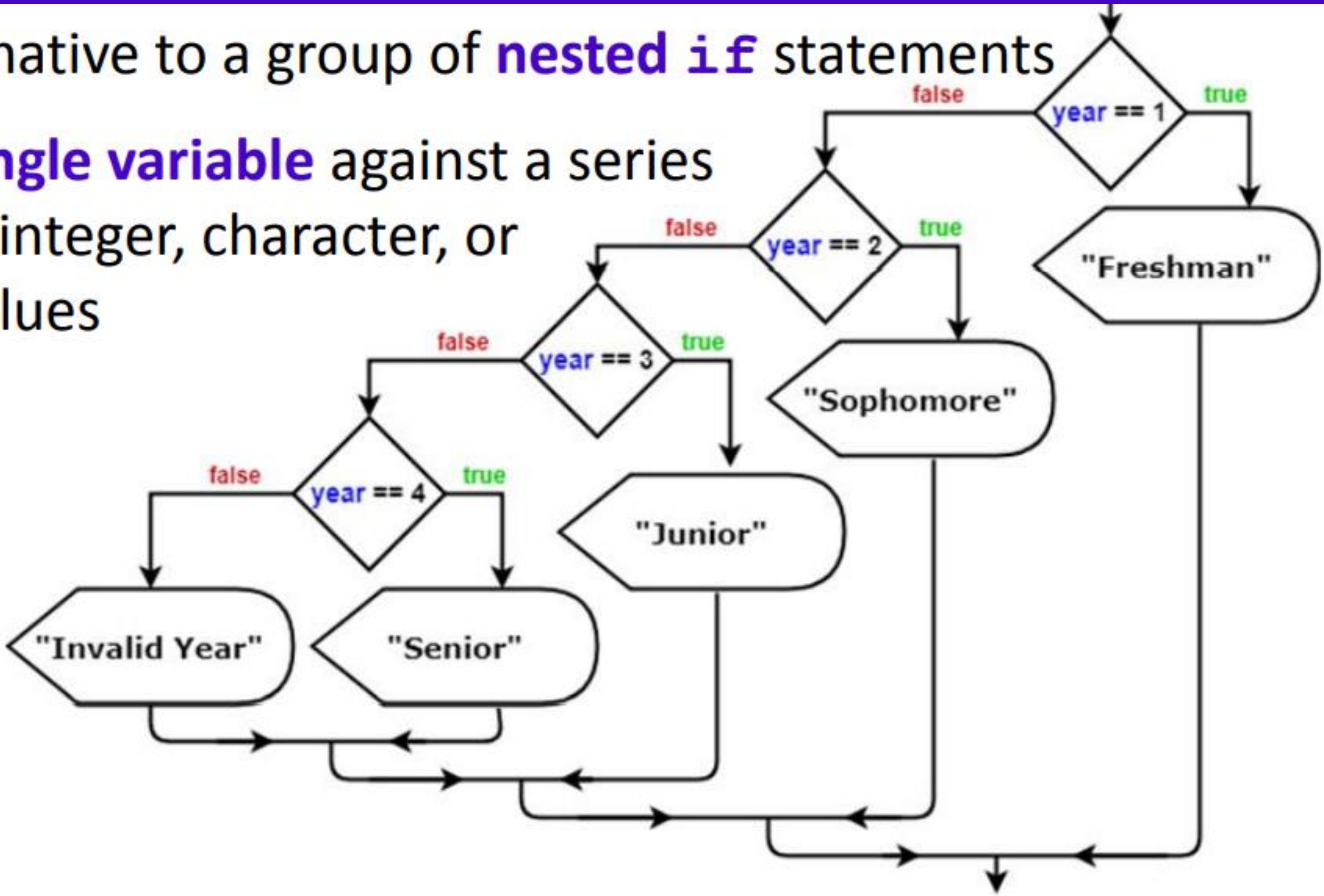
Suppose score is 70.0

Exit the if statement

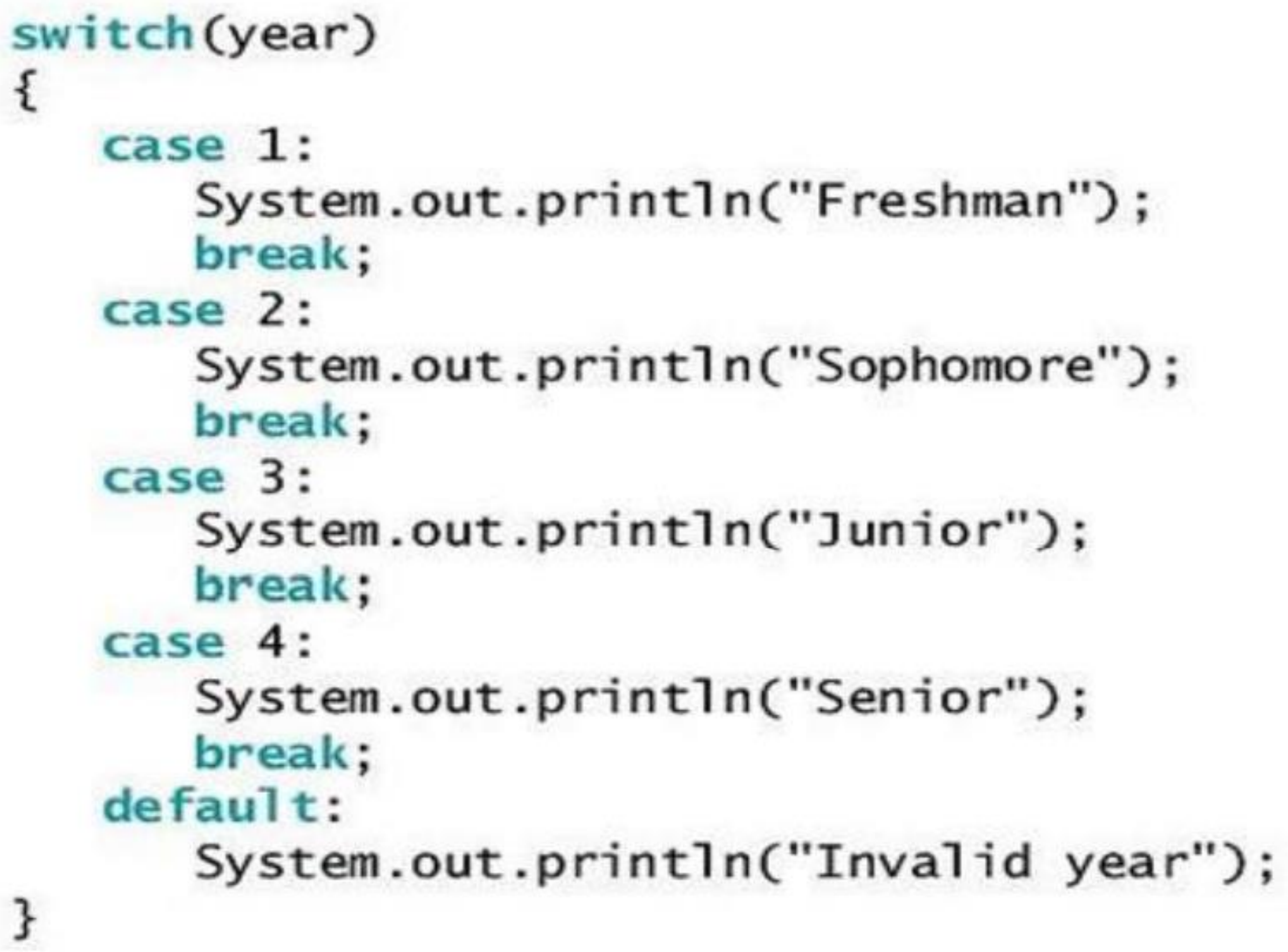
```
if (score >= 90.0)
  System.out.print("A");
else if (score >= 80.0)
  System.out.print("B");
else if (score >= 70.0)
  System.out.print("C");
else if (score >= 60.0)
  System.out.print("D");
else
  System.out.print("F");
```

switch Statement

- An alternative to a group of **nested `if`** statements
- Test a **single variable** against a series of exact integer, character, or string values



Example - switch Statement



```
switch(year)
{
    case 1:
        System.out.println("Freshman");
        break;
    case 2:
        System.out.println("Sophomore");
        break;
    case 3:
        System.out.println("Junior");
        break;
    case 4:
        System.out.println("Senior");
        break;
    default:
        System.out.println("Invalid year");
}
```

switch Statement and keywords

- switch: Starts the structure and Followed by a test expression enclosed in parentheses
- case: Followed by one of the possible values for the test expression and a colon
- break: Optionally terminates a switch statement at the end of each case
- default: Optionally is used prior to any action that should occur if the test variable does not match any case

Example of switch-case fall-through

Example – Is num a multiple of 4?

```
rem = num % 4;
switch (rem) {
    case 0:
          System.out.print(num + " i s a multiple of 4 . " ) ;
          break;
    case 1:
    case 2:
    case 3:
          System.out.print(num + " i s not a multiple of 4 . " ) ;
} // end switch
```

- Program falls through cases until hitting break; or closing brace
- Could just have `case 0` and `default` here instead

More Practice – Grading using switch

- Rewrite the grading program using switch statement.