



COSC 111

Computer Programming I

Chapter 3 Selections

Dr. Abdallah Mohamed

Selection Example

When you use 'selection', the program can decide which statements to execute based on a condition.

```
double radius = 10, area;  
if (radius < 0) {  
    System.out.println("Incorrect value");  
} else {  
    area = radius * radius * 3.14159;  
    System.out.println("Area is " + area);  
}
```

Selection statements use **conditions that are Boolean expressions.**

Relational Operators and Boolean Type

Relational Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
<	<	less than	<code>radius < 0</code>	<code>false</code>
<=	≤	less than or equal to	<code>radius <= 0</code>	<code>false</code>
>	>	greater than	<code>radius > 0</code>	<code>true</code>
>=	≥	greater than or equal to	<code>radius >= 0</code>	<code>true</code>
==	=	equal to	<code>radius == 0</code>	<code>false</code>
!=	≠	not equal to	<code>radius != 0</code>	<code>true</code>

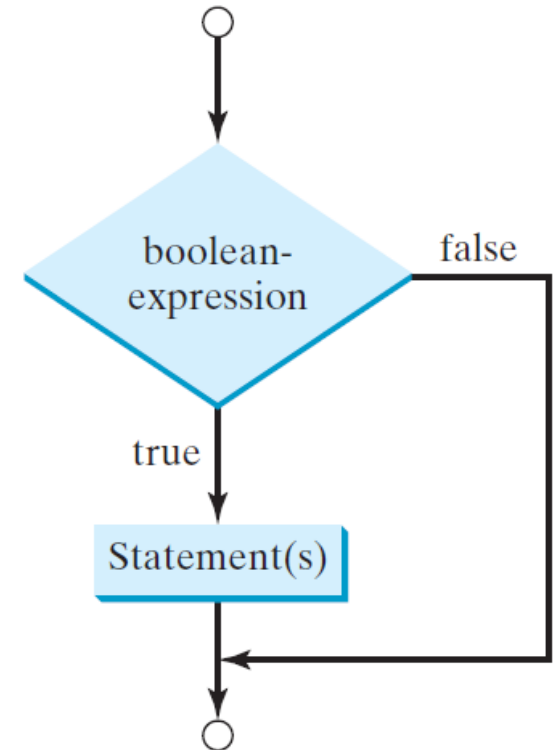
Example:

```
boolean b = (1 < 2);
```

```
System.out.println(b); // displays true
```

One-way if Statements

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

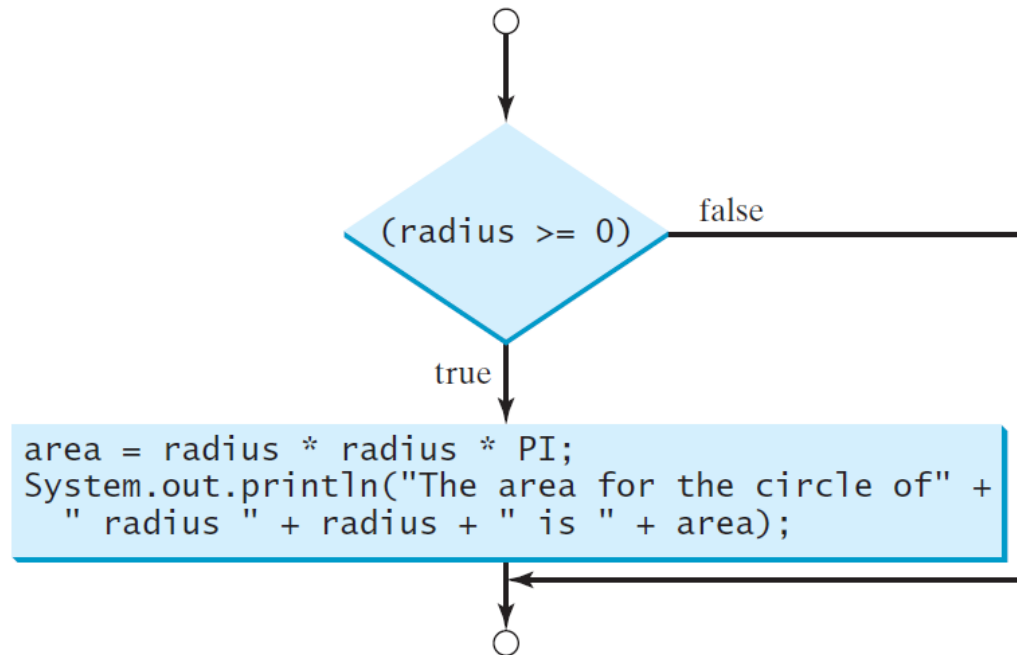


Note: The block braces { } can be omitted if they enclose a single statement.

One-way if Statements

Example:

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



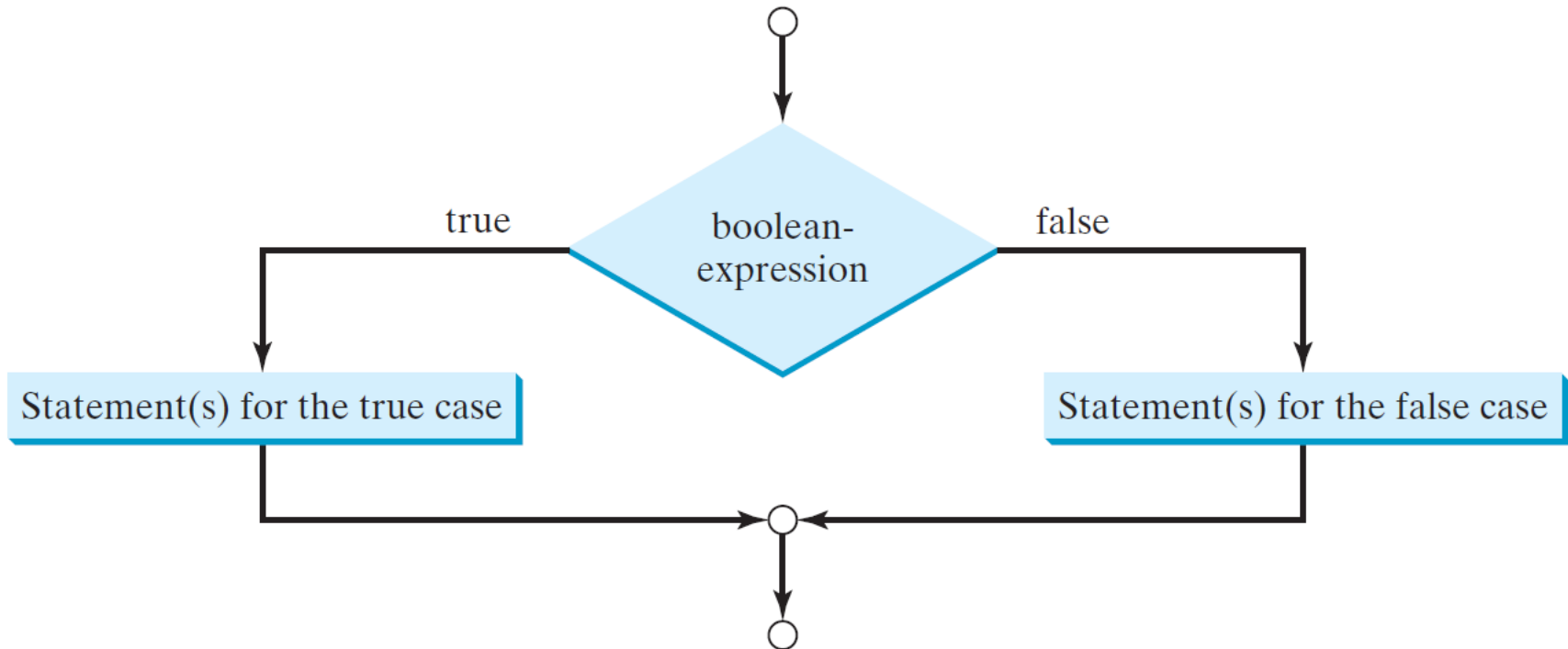


Write a program that prompts the user to enter an integer.

- If the number is a multiple of 5, print HiFive.
- If the number is divisible by 2, print HiEven.
- That is, you could have one of the following three outputs:
 - HiFive
 - HiEven
 - HiFive HiEven

Two-way if Statement

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



Two-way if Statement: Example

```
double radius = 10, area;  
if (radius < 0) {  
    System.out.println("Incorrect value");  
} else {  
    area = radius * radius * 3.14159;  
    System.out.println("Area is " + area);  
}
```


Clicker Question

What is the output?

```
int x = 10;  
if (x <= 10)  
    System.out.print("A") ;  
else  
    System.out.print("B") ;  
    System.out.print("C") ;
```

A. A

B. B

C. ABC

D. AB

E. AC

Clicker Question

What is the output?

```
int x = 10;  
if (x <= 10)  
    System.out.print("A") ;  
else{  
    System.out.print("B") ;  
    System.out.print("C") ;  
}
```

A. A

B. B

C. ABC

D. AB

E. AC

Practice

- 1) Write an **if** statement that assigns “**positive**” to **String** type if **int n** is greater than **0**.

- 2) Write an **if** statement that prints out “passed” if grade is larger than or equal to 50, otherwise print out “failed”.

- 3) Write an **if** statement that increases **double pay** by 3% if **score** is greater than **90**

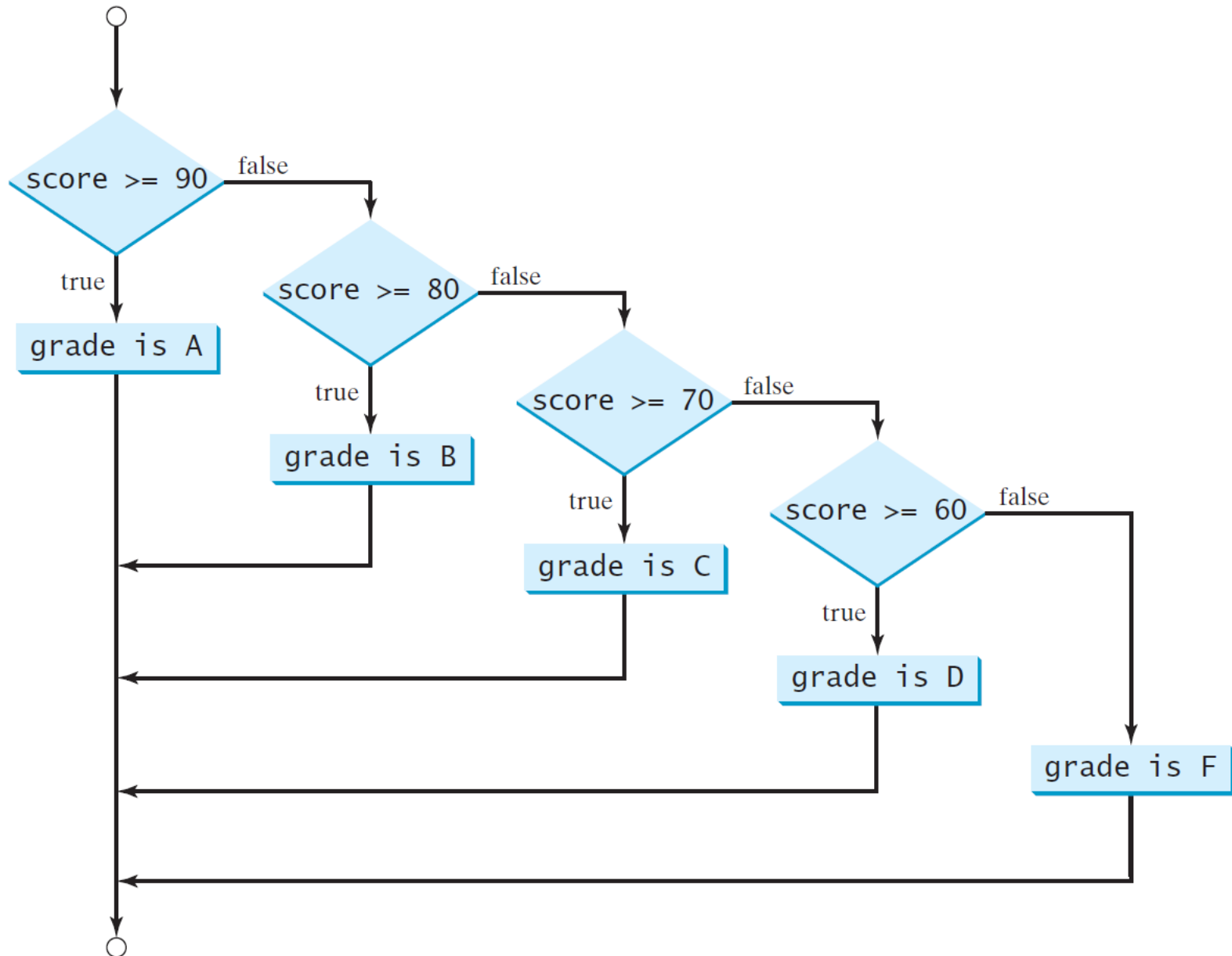
Nesting if Statement

```
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");
```

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");
```

Nesting if Statement



TIPS

(1) Forgetting Necessary Braces

- The braces can ONLY be omitted if the block contains a single statement.

```
double radius = 5, area;  
if (radius >= 0)  
    area = radius * radius * 3.14;  
    System.out.println("The area " + " is " + area);  
  
//WRONG!
```

```
double radius = 5, area;  
if (radius >= 0){  
    area = radius * radius * 3.14;  
    System.out.println("The area " + " is " + area);  
}  
  
//CORRECT!
```

TIPS

(2) Wrong Semicolon at the if Line (Logic Error)

- The following two code segments are equivalent.

```
if (radius >= 0);    //LOGICAL ERROR
{
    area = radius * radius * 3.14;
    System.out.println("The area " + " is " + area);
}
```

```
double radius = 5, area;
if (radius >= 0) {};
{
    area = radius * radius * 3.14;
    System.out.println("The area " + " is " + area);
}
```

(3) Dangling else Ambiguity

```
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)

TIPS

(4) Avoiding Duplicate Code in Different Cases

- The following two code segments are equivalent.

```
if (inState) {  
    tuition = 5000;  
    System.out.println("The tuition is " + tuition);  
} else {  
    tuition = 15000;  
    System.out.println("The tuition is " + tuition);  
}
```

```
if (inState) {  
    tuition = 5000;  
} else {  
    tuition = 15000;  
}  
System.out.println("The tuition is " + tuition);
```

TIPS

(5) Redundant Testing of Boolean Values

- The following two code segments are equivalent.

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

(a)

Equivalent

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

(b)

Clicker Question

What is the output?

```
int num=12;  
if (num >= 8)  
    System.out.print("A");  
    if (num == 10)  
        System.out.print("B");  
else  
    System.out.print("C");
```

A. A

B. B

C. C

D. AB

E. AC



Create a program to teach a first grade child how to learn subtractions. The program **randomly** generates two single-digit integers number1 and number2 **with number1 \geq number2** and displays a question such as “What is 9 – 2?” to the student. After the student types the answer, the program displays whether the answer is correct.

- **Hint:** *use `Math.random()` to obtain a random double value between 0 and 9 inclusive.*

Algorithm

1. Generate two random single-digit integers
2. Make sure number 1 is larger than number (**How?**:
if number1 < number2, swap number1 with number2)
3. Get student's answer to the question: “what is number1 – number2?”
4. Grade the answer and display the result



Write a program that prompts the user to enter a weight in kilograms and height in centimeters, then displays the BMI.

- Body Mass Index (BMI) is a measure of health on weight:
 - $\text{BMI} = \text{weight in kilograms} / (\text{height in meters})^2$.
 - The interpretation of BMI for people 16 years or older is as follows:

BMI	Interpretation
$\text{BMI} < 18.5$	Underweight
$18.5 \leq \text{BMI} < 25.0$	Normal
$25.0 \leq \text{BMI} < 30.0$	Overweight
$30.0 \leq \text{BMI}$	Obese

Algorithm?

1. Prompt the user to enter weight in pounds & height in inches
2. Compute BMI
3. Display result based on calculated BMI value

Logical Operators

*The logical operators **!**, **&&**, **||**, and **^** can be used to create a **compound Boolean expression**.*

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
 	or	logical disjunction
^	exclusive or	logical exclusion

NOTE: In Java, the expression **1 <= numberOfDaysInAMonth <= 31** is **INCORRECT**, because **1 <= numberOfDaysInAMonth** is evaluated to a **boolean** value, which cannot be compared with **31**. The correct expression in Java is

(1 <= numberOfDaysInAMonth) && (numberOfDaysInAMonth <= 31)

Truth Tables for Logical Operators

p	!p
true	false
false	true

p ₁	p ₂	p ₁ && p ₂	p1 p2	p1 ^ p2
false	false	false	false	false
false	true	false	true	true
true	false	false	true	true
true	true	true	true	false

Practice

Let's say a person can get a driving license if s/he satisfies exactly two conditions:

- To be at least 17 years old.
- To currently be living in BC.

1) Write a Java program to check the **age** (answer is an **integer**) and ask a Y/N question about the living address (answer is **character**). The program should inform the user whether they can get a driving license or not.

2) display to the user why they were denied a driving license (if applicable)

Clicker Question

What is the output?

```
int x = 10, y = 20;  
boolean result = !(x != 10) && (y == 20);  
System.out.println(result);
```

A. true

B. false

Clicker Question

What is the output?

```
int x = 10, y = 20;  
boolean result = (x >= y) || (y <= x);  
System.out.println(result);
```

A. true

B. false

Clicker Question

What is the output?

```
int x = 10, y = 20;
if (x >= 5) {
    System.out.print("bigx ");
    if (y >= 10)
        System.out.print("bigy ");
} else if (x == 10 || y == 15)
    if (x < y && x != y)
        System.out.print("not equal");
```

A. bigx

B. bigy

C. bigx not equal

D. bigx bigy not equal

E. bigx bigy



Write a program that prompts the user to enter a number and then checks whether that number is divisible by 2 **and** 3, whether a number is divisible by 2 **or** 3, and whether a number is divisible by 2 **or** 3 **but not both**.

Algorithm:

1. Prompt the user to enter a number
2. Check the number and display the result

Hint: use AND (&), OR (|), and XOR (^) operators

The & and | Operators

&& and ||:

- Shortcut operators
- The parts of an expression containing && or || operators are evaluated only until it's known whether the condition is true or false .

& and |

- the & and | operators **always evaluate both of their operands.**

Practice

■ What is the value of x after these expressions?

- `int x = 1;`
`if ((x > 1) & (x++ < 10));` `//x = 2`

- `int x = 1;`
`if ((x > 1) && (x++ < 10));` `//x = 1`

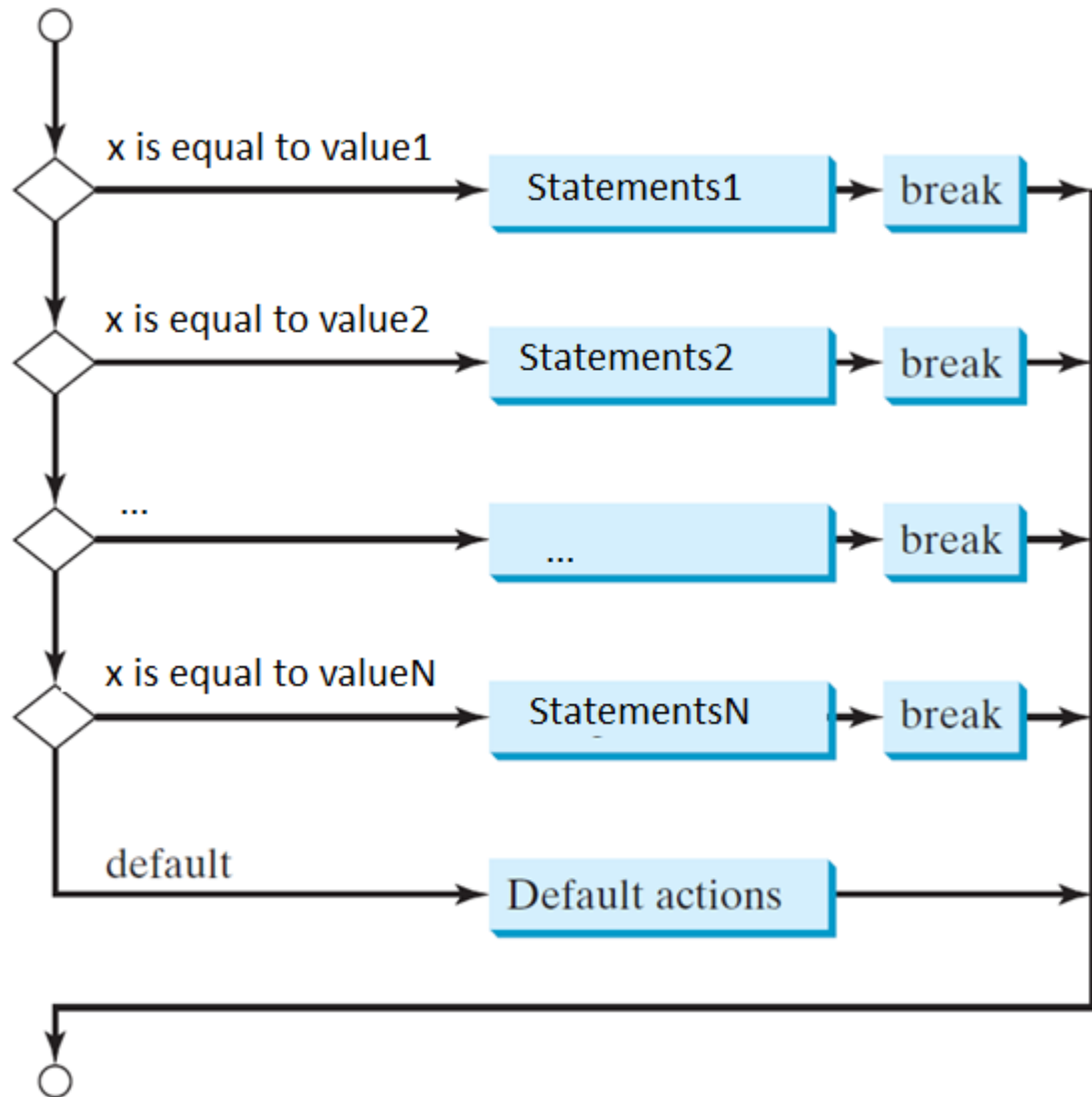
- `int x = 1;`
`if ((1 == x) | (10 > x++));` `//x = 2`

- `int x = 1;`
`if ((1 != x) || (10 > x++));` `//x = 2`

Switch Statement

switch Statement

```
switch (x) {  
    case value1:  
        //statements1;  
        break;  
    case value2:  
        //statements2;  
        break;  
    ...  
    case valueN:  
        //statementsN;  
        break;  
    default:  
        //default actions;  
}
```



switch Statement

The switch-expression (x in the figure) :

- must yield a value of char, byte, short, or int type (also String in JDK7 and beyond)
- must always be enclosed in parentheses.

The values:

- value1, ..., and valueN must have the same data type as the value of the switch-expression.

The case statements

- are executed when the value in the case statement matches the value of the switch-expression.

The keyword **break**:

- It is optional, but it should be used at the end of each case in order to terminate the remainder of the switch statement. If the break statement is not present, the next case statement will be executed.

The default case

- It is optional, and it can be used to perform actions when none of the specified cases matches the switch-expression.

```
switch (x) {  
    case value1:  
        //statements1;  
        break;  
    case value2:  
        //statements2;  
        break;  
    ...  
    case valueN:  
        //statementsN;  
        break;  
    default:  
        //default actions;  
}
```

Clicker Question

What is the output?

```
int num = 2;  
switch (num) {  
    case 2: System.out.print("two "); break;  
    case 1: System.out.print("one "); break;  
    case 3: System.out.print("three "); break;  
    default: System.out.print("other "); break;  
}
```

- A. one
- B. two
- C. three
- D. other

Clicker Question

What is the output?

```
int num = 2;  
switch (num) {  
    case 1:  System.out.print("one ");  
    case 3:  System.out.print("three "); break;  
    case 2:  System.out.print("two ");  
    default: System.out.print("other ");  
}
```

- A. one
- B. one two three
- C. one three two other
- D. two other
- E. one three

Clicker Question

What is the output?

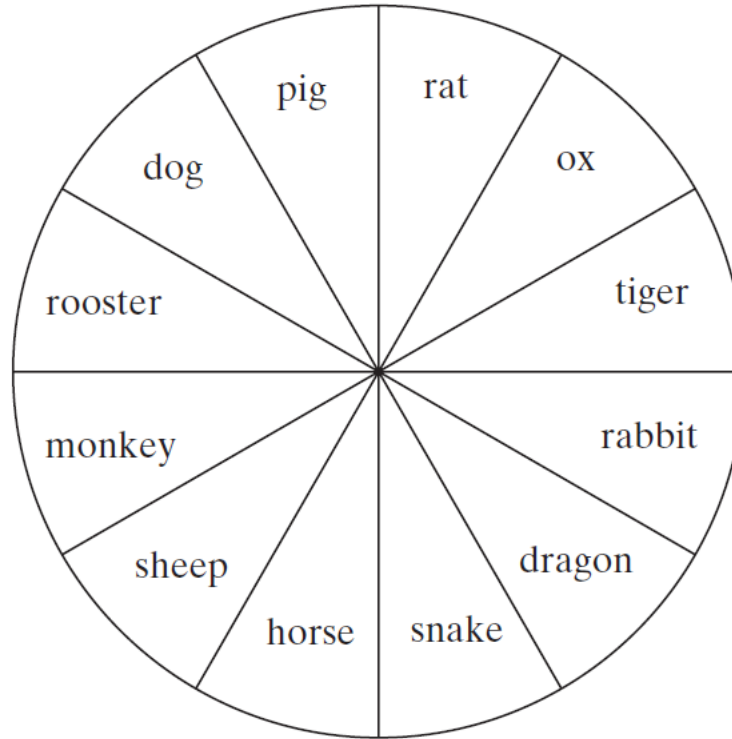
```
int day = 2;  
switch(day) {  
    case 1:  
    case 2:  
    case 3:  
    case 4:  
    case 5: System.out.print("weekday"); break;  
    case 0:  
    case 6: System.out.print("weekend"); break;  
    default: System.out.print("unknown day");  
}
```

- A. weekday
- B. weekend
- C. unknown day
- D. None of the above

Practice



Write a program that prompts the user to enter a year and displays the animal for the year.



$\text{year} \% 12 =$ {
0: monkey
1: rooster
2: dog
3: pig
4: rat
5: ox
6: tiger
7: rabbit
8: dragon
9: snake
10: horse
11: sheep

Algorithm:

1. Prompt the user to enter a year
2. Calculate $\text{year} \% 12$
3. Use conditional statements to display the correct animal.

Conditional Expression (? :)

Conditional Expressions

A conditional expression evaluates an expression based on a condition. Its syntax is:

boolean-expression ? expression1 : expression2;

Example1:

```
y = (x > 0) ? 1 : -1;
```

is equivalent to:

```
if (x > 0)
    y = 1;
else
    y = -1;
```

Conditional Expressions

Example2:

```
System.out.println((num % 2 == 0) ? num + "is even" : num + "is odd");
```

is equivalent to:

```
if (num % 2 == 0)
    System.out.println(num + "is even");
else
    System.out.println(num + "is odd");
```


TIPS

Simplifying Boolean Variable Assignment

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

(a)

Equivalent

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

(b)

Clicker Question

What is the displayed on the screen?

```
int x = 5, y = 2;  
x > y? "larger": "smaller";
```

- A. larger
- B. smaler
- C. larger smaller
- D. Error

Clicker Question

What is the displayed on the screen?

```
int x = 5, y = 2;  
String s = x > y? "larger": "smaller";  
System.out.println(s);
```

- A. larger
- B. smaler
- C. larger smaller
- D. Error

Practice

Rewrite the following **if** statements using the conditional operator.

```
if (x % 2 == 0)
```

```
    s = "even";
```

```
else
```

```
    s = "odd";
```

```
if (y >= 10)
```

```
    x = 100;
```

```
else
```

```
    x = -100;
```

Practice

Rewrite the following conditional expressions using **if-else** statements


- `score = (x > 10) ? 3 * scale : 4 * scale;`
- `System.out.println((number % 3 == 0) ? i : j);`

Write conditional expression that returns **-1** or **1** randomly.

Operator Precedence

When two operators share an operand, the operator with the higher *precedence* goes first.

- For example, $1 + 2 * 3$ is treated as $1 + (2 * 3)$ since multiplication has a higher precedence than addition.



<code>var++</code> and <code>var--</code> (Postfix) <code>+</code> , <code>-</code> (Unary plus and minus), <code>++var</code> and <code>--var</code> (Prefix) (type) (Casting) <code>!</code> (Not)	Unary Operations
<code>*</code> , <code>/</code> , <code>%</code> (Multiplication, division, and remainder) <code>+</code> , <code>-</code> (Binary addition and subtraction)	Mathematical Operators
<code><</code> , <code><=</code> , <code>></code> , <code>>=</code> (Relational) <code>==</code> , <code>!=</code> (Equality)	Relational Operators
<code>^</code> (Exclusive OR) <code>&&</code> (AND) <code> </code> (OR)	Logical Operators
<code>=</code> , <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> (Assignment operator)	Assignment

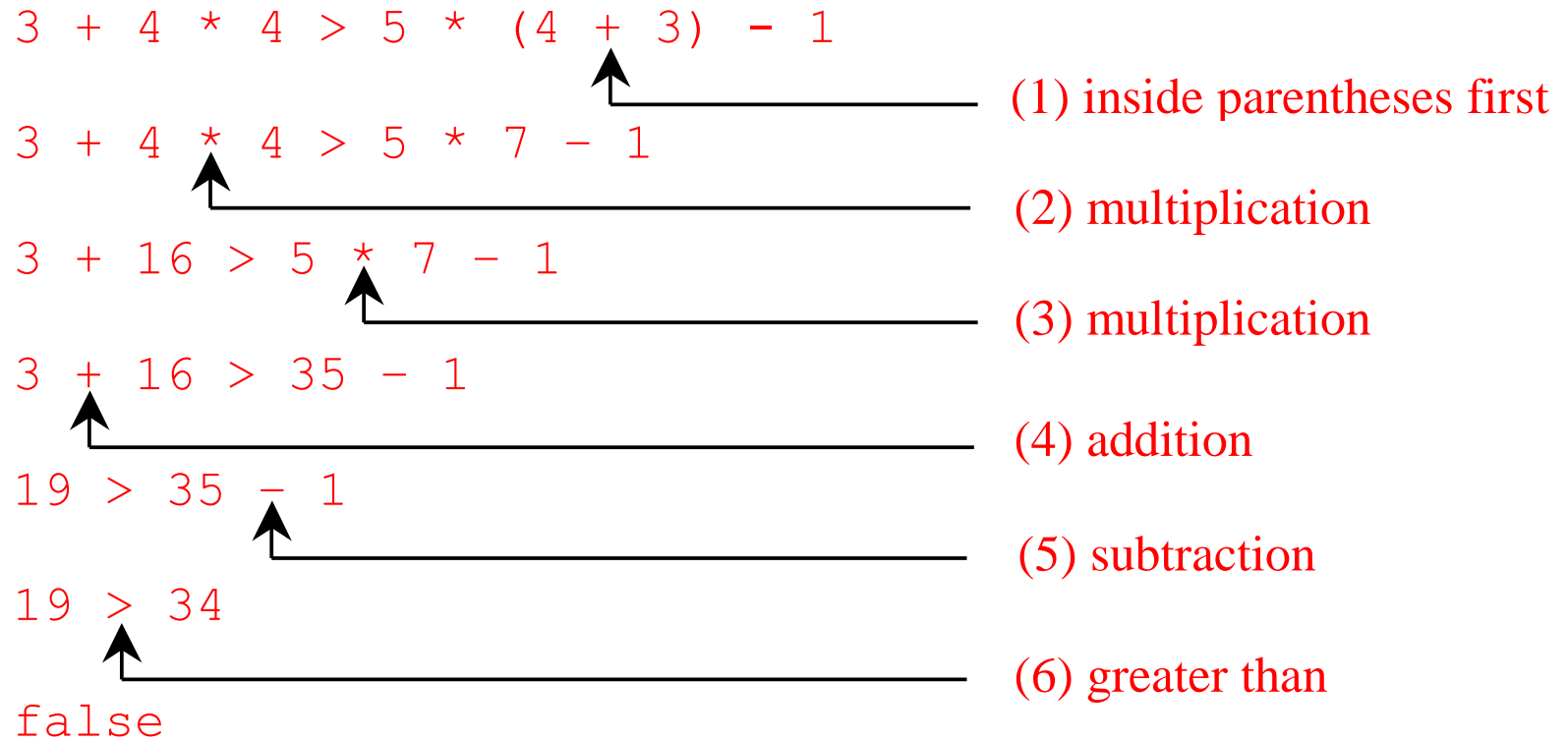
Operator Precedence and Associativity

Notes:

- The expression in the **parentheses is always evaluated first**.
When evaluating an expression without parentheses, the operators are applied according to the precedence rule and the associativity rule.
- If operators with the same precedence are next to each other, their associativity determines the order of evaluation.
 - **All binary operators** except assignment operators are **left-associative**.
 $a - b + c - d$ is equivalent to $((a - b) + c) - d$
 - **Assignment operators** are **right-associative**.
 $a = b += c = 5$ is equivalent to $a = (b += (c = 5))$

Example

Applying the operator precedence and associativity rule, the expression $3 + 4 * 4 > 5 * (4 + 3) - 1$ is evaluated as follows:



Practice

What is the value of this expression?

- $3 + 4 * 4 > 5 * (4 + 3) - 1 \ \&\& \ (4 + 3 > 5) ;$

Answer: apply the precedence rules as follows:

- Parentheses $3 + 4 * 4 > 5 * 7 - 1 \ \&\& \ (4 + 3 > 5)$

$3 + 4 * 4 > 5 * 7 - 1 \ \&\& \ (7 > 5)$

$3 + 4 * 4 > 5 * 7 - 1 \ \&\& \ \text{true}$

- Mathematical $19 > 34 \ \&\& \ \text{true}$

- Relational $\text{false} \ \&\& \ \text{true}$

- Logical false

Practice

Assuming that **x** is **1**, what is the value of.

- **(true) && (3 > 4)**
- **!(x > 0) && (x > 0)**
- **(x > 0) || (x < 0)**
- **(x != 0) || (x == 0)**
- **(x >= 0) || (x < 0)**
- **(x != 1) == !(x == 1)**

Debugging

Remember: Programming Errors

3 types of errors:

■ Syntax Errors

- Detected by the compiler
- aka *compilation errors*

```
public class Errors {  
    public static main(String[] args) {  
        System.out.println("Welcome to Java");  
    }  
}
```

■ Runtime Errors

- Causes the program to abort during the runtime.

```
public class Errors {  
    public static void main(String[] args) {  
        System.out.println(1 / 0);  
    }  
}
```

Can't divide by zero

■ Logic Errors

- Produces incorrect result during the runtime
- no error message is shown

```
public class Errors {  
    public static void main(String[] args) {  
        System.out.println("35 Celsius in Fahrenheit:");  
        System.out.println((9 / 5) * 35 + 32);  
    }  
}
```

Output is incorrect
due to wrong formula

Debugging your code

Debugging is the act of finding and correcting errors in a system.

A common **reason for computer errors** is our **lack of precision** in specifying instructions to the computer

As a programmer, you need to know how to debug your code.

Eclipse provides us with tools to help us identify the source of errors our code.

Both Syntax and Runtime errors are easily found whenever they occur (with the help with the error message that appears on the console).

- Syntax errors are identified before compilation.
- Runtime errors are identified while the program is running.

Logic errors can be located using Eclipse Debugger

Remember: Eclipse: Debugging and Breakpoints

Debug button

Step and Play Buttons

Breakpoint

Code editor

Variable view

Next statement to execute

Console (execution)

Java - Debug - COSC111 Projects/src/HelloWorld.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Console

Variables

Breakpoints

Outline

Code Editor

```
1  
2 public class HelloWorld {  
3  
4     public static void main(String[] args) {  
5         System.out.println("Hello World!");  
6         System.out.println("Bye");  
7     }  
8
```

Console

Tasks

HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (Sep 10, 2017, 1:08:20 PM)

Hello World!

Writable Smart Insert 6:1