# COP-2210
# Computer Programming I

Instructor: Dr. Antonio Hernandez

Text: Big Java: Early Objects, Interactive Edition, 6th Edition

# The Java Language

## 6. Application Programs

## General structure of a Java program

A ***Java*** program is a collection of classes

Definition (or *declaration*) of a class:

```
<modifiers> class <name of the class>
{
   // Declaration of variables and methods
}
```

## Variables

***Variable***: Where data are stored.

Declaration of a *variable*:

<variable type> <*name of the variable*> = <value>;

## Structure of a *method*

***Method***: a sequence of instructions to perform a task.

Definition (or *declaration*) of a method:

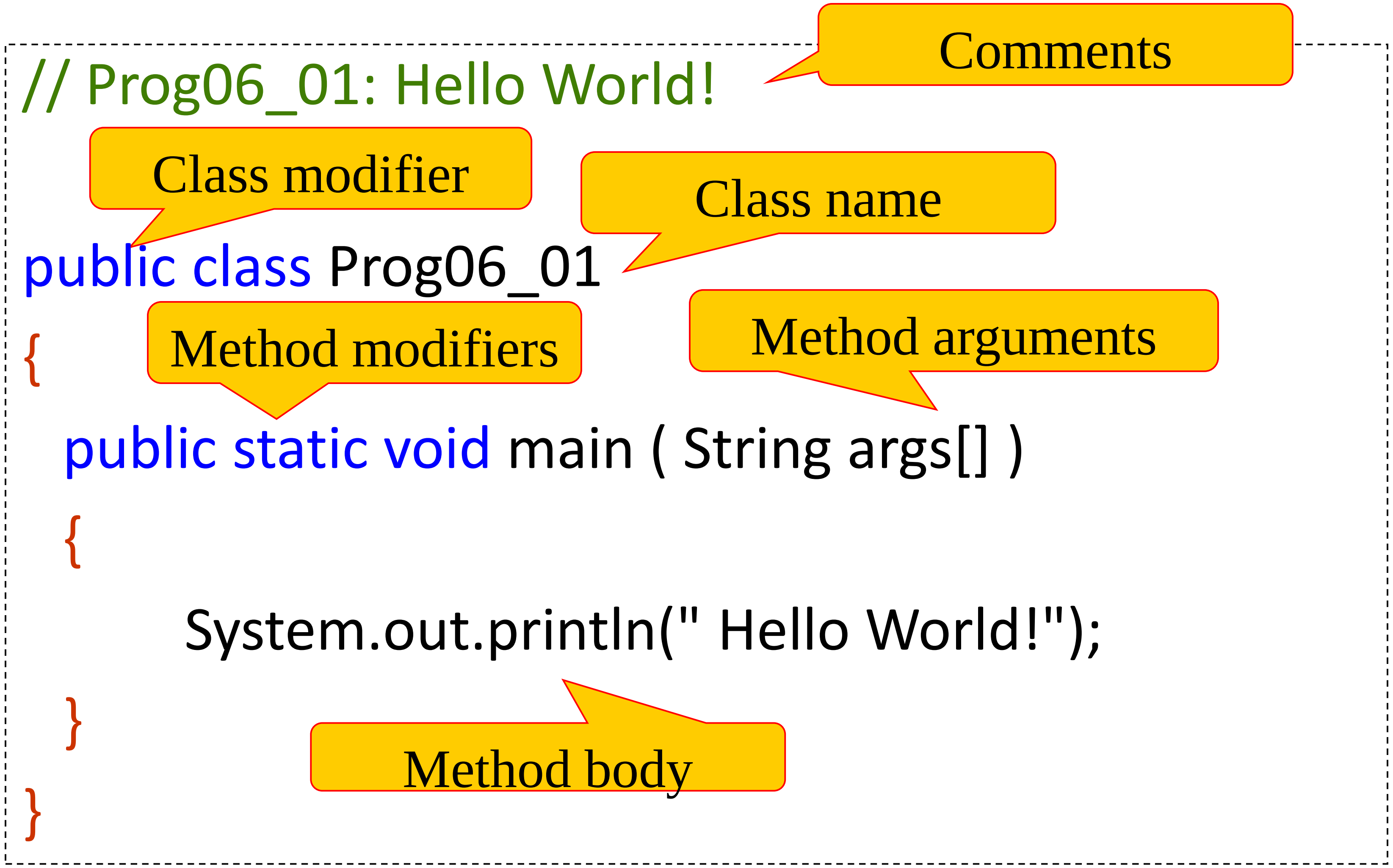
```
<modifiers> <return type> <name of the method>( <arguments> )
{
   // method body. Variable declarations might be included also.
}
```

## Example: “Hello World” program

```
// Prog06_01: Hello World!


public class Prog06_01
{
  public static void main ( String args[] )
  {
        System.out.println(" Hello World!");
  }
}
```

# “Hello World” Program Elements



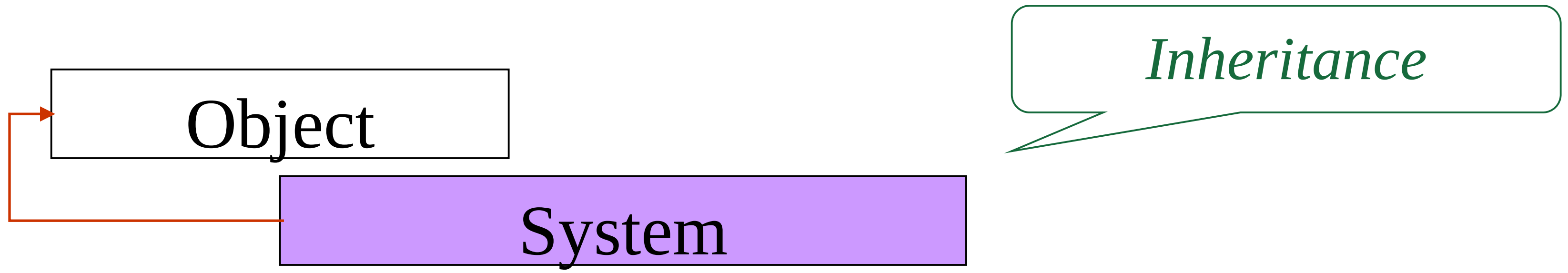
```
// Prog06_01: Hello World!

public class Prog06_01
{
   public static void main ( String args[] )
   {
         System.out.println(" Hello World!");
   }
}
```

# The *System* class

The ***System*** class contains several useful class fields and methods.

Facilities provided by the System class: *standard input, standard output*, and *error output streams.*



## The *System* class

***System.out***: The "standard" output device (video screen for most systems).

***System.out.println(“ . . . ”);***
***System.out.print(“ . . . ”);***

Displays the string of characters “ . . . ” on the standard output display device

println(): new line is added
print(): no new line

# Example

```
/* Prog06_02: New Hello World! -- and more
  than one line of comment -- */

public class Prog06_02
{
  public static void main(String args[])
  {
        System.out.println ( );
        System.out.print (" Hello ");
        System.out.println ("World!");
  }
}
```

## Example

```
//Prog06_03 Practice with print statements


public class Prog06_03
{
   public static void main ( String args [ ] )
   {
         System.out.println ( "1 + 2" );
         System.out.println ( 1 + 2);
   }
}
```

## Whitespaces

```
/* Prog06_02: New Hello World! -- and more
   than one line of comment -- */


public class Prog06_02 { public static void main(String
   args[]) { System.out.println(); System.out.print(" Hello
   "); System.out.println(“World!”); } }
```

## Writing Styles: the good …

```
/* Prog06_02: New Hello World! -- and more
  than one line of comment -- */


public class Prog06_02
{
  public static void main(String args[])
  {
        System.out.println ( );
        System.out.print (" Hello ");
        System.out.println ("World!");
  }
}
```

## Escape Sequences

| Escape sequence | Purpose |
|---|---|
| \n | New line (screen cursor goes to next line) |
| \r | Carriage return (screen cursor goes to the beginning of current line |
| \t | Tab |
| \\ | Backslash |
| \” | Double quote |

# The Java Language

## 7. Data Types

## Data Types

**Data Type**: A type of "container" that can hold a specific kind of program data

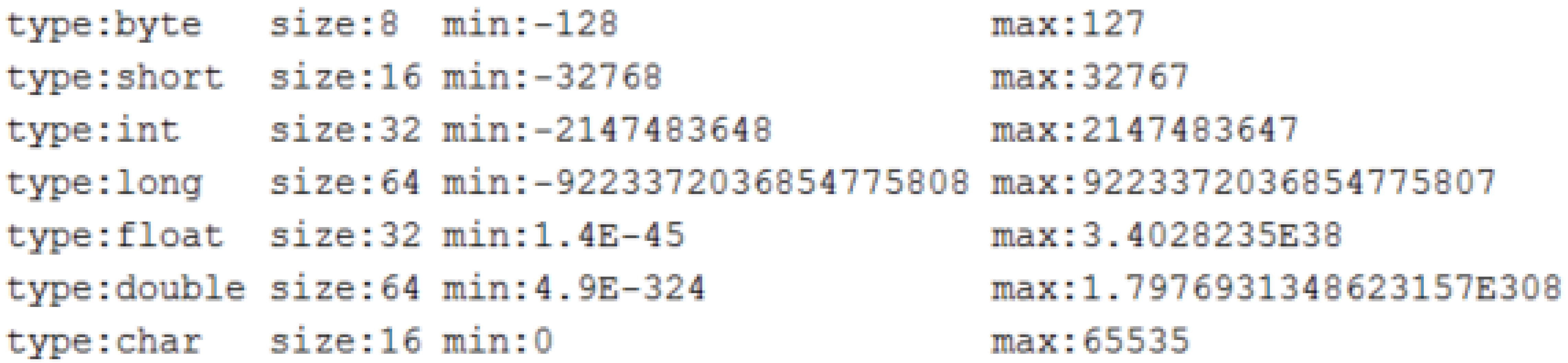
***Basic or Primitive Data Types***

| | | | |
|---|---|---|---|
| *byte* | *short* | *int* | *long* |
| *float* | *double* | | |
| *char* | | | |
| *boolean* | | | |

## Primitive Data Types

| Data Type | Bytes | Data it contains |
|---|---|---|
| byte | 1 | integer |
| short | 2 | integer |
| int | 4 | integer |
| long | 8 | integer |
| float | 4 | real |
| double | 8 | real |
| boolean | 1 | *true*, *false* |
| char | 2 | character |

# Primitive Data Types



```
type:byte   size:8  min:-128                 max:127
type:short  size:16 min:-32768               max:32767
type:int    size:32 min:-2147483648          max:2147483647
type:long   size:64 min:-9223372036854775808 max:9223372036854775807
type:float  size:32 min:1.4E-45              max:3.4028235E38
type:double size:64 min:4.9E-324             max:1.7976931348623157E308
type:char   size:16 min:0                    max:65535
```

# The Java Language

## 8. Variables

## Variables

**Variable**: The actual location in memory
set aside for use by the program

- Variables contain values that may be modified during the execution of a program.

*(variables could be think of as the math variables)*

## Variables: declaration

Variables must be declared:

**<variable type> <*name of the variable*> = <value>**

**Ex.**

```
float balance;
double deposit = 1000.0;
int transaction_count;
int check_number = 421;
```

# Variables: assigning value

The assignment operator:

**<variable name>** = **<expression>**

The ***equals sign***, called the assignment operator, takes the value on the right side and places it in the variable on the left side.

**Ex**.

```
x = 12.345;
```

## Variables: *Try it Yourself*

```
// Prog08_01: Declaring variables in Java

public class Prog08_01
{
   public static void main ( String args[] )
   {
         double d;
         d = -2.56;
         System.out.print ( “This is d: ” );
         System.out.println ( d );
   }
}
```

## Variables: *Try it Yourself*

```
// Prog08_02 Declaring variables in Java

public class Prog08_02
{
  public static void main ( String args[ ] )
  {
        double d;
        d = -2.56;
        System.out.println ( “This is d: ” + d );
  }
}
```

## PRACTICE

- Edit and compile the *Program 08_02*

- In *Program 08_02*:

  ► Comment the line ***double d;***
  What did it happen?

  ► Uncomment the line. Replace ***double d;*** with ***char d;***
  What did it happen?

# Variables: Declaration

```
// Prog08_01: Declaring variables in Java

public class Prog08_01
{
  public static void main ( String args[] )
  {
        double d;              ←———————  CORRECT
        d = -2.56;
        System.out.print(“This is d: ”);
        System.out.println(d);
        double d;              ←———————  INCORRECT
  }
```

Java requires that a variable be declared **before** its is used.

## Variables: Naming Rules

1. Names may contain letters, numbers, underscores (_) or a dollar sign ($)
2. The first character must be a letter, an underscore or a dollar sign
3. Names cannot contain any symbols, such as

   ~!@#%^&*()-+=\|/

   nor can they have spaces.
4. Keywords cannot be used as variable names.
5. Variables names may be any length

*These rules must be followed when forming any identifier (user-defined classes, methods, packages)*

## Variables: Naming Rules

| **Variable Name** | |
|---|---|
| *i* | Valid |
| *new_VAR* | Valid |
| *int* | Invalid |
| *3d* | Invalid |
| *price#* | Invalid |
| $price | Valid |

# PRACTICE

- Correct the errors:

```
\\ Prog08_03 Declaring variables in Java
public class Prog08_03
{
    public void static main ( String args[] )
    {
        i& = 3;
        int i&;
        system.out.print (i&, " is the value assigned to i&/n/n");
    }
```

# PRACTICE

- Correct the errors:

```
// Prog08_03 Declaring variables in Java
public class Prog08_03
{
    public static void main ( String args[] )
    {
        int i;
        i = 3;
        System.out.print (i + " is the value assigned to i\n\n");
    }
}
```

## Modifiers: *final*

**Final variable: Used to fix a variable’s value forever.**

(It is an error to change its value)

**final <type> <variable name> = <initial value>;**

**Example**

```
final int x = - 5;
```

## *String* Variables and Expressions

**STRING class and concatenation**

```
public class Prog08_04
{
   public static void main( String args[ ])
   {
         String firstName = "Joan";
         String lastName = "Smith";
         String output = "Name: " + firstName + " " + lastName;

         System.out.println(output);
   }
}
```

## *String* Variables and Expressions

**STRING class and concatenation**

```
public class Prog08_05
{
  public static void main( String args[ ])
  {
        String s;
        double Pi = 3.14;
        int dec = 92;
        s="PI= " + Pi + 15 + dec + "...";
        System.out.println(s);
  }
}
```

# The Java Language

## 9. Input From the Keyboard

## Input from the keyboard: using the Scanner class

```
import java.util.*;

public class Prog09_01
{
    public static void main ( String args[ ] )
    {
            Scanner in  =  new Scanner ( System.in );

            System.out.print ( "Enter INT: " );
            int x  =  in.nextInt ( );
            System.out.println ( "You entered: " + x );

            System.out.print ( "Enter DOUBLE: " );
            double y  =  in.nextDouble ( );
            System.out.println ( "You entered: " + y );
    }
}
```

## PRACTICE

**Program 09_02**

Write a *Java* program that:

a) asks the user to enter his/her *height* (a decimal number).
b) after reading the height, the program will display it.

## PRACTICE - ANSWER

```java
import java.util.*;

public class Prog09_02
{
   public static void main(String args[])
   {
      Scanner in = new Scanner(System.in);

      System.out.print("Enter HEIGHT: ");
      double height = in.nextDouble();

      System.out.println("Your height is: " + height);
   }
}
```

# Input from the keyboard: using the Scanner class

```java
import java.util.*;

public class Prog09_03 {
    public static void main ( String args[ ] ) {
            Scanner in  =  new Scanner ( System.in );

            System.out.print ( "Enter STRING: " );
            String s  =  in.nextLine ( );
            System.out.println ( "You entered: " + s);

            System.out.print ( "Enter FLOAT: " );
            float f  =  in.nextFloat( );
            System.out.println ( "You entered: " + f );
```

```java
            System.out.print ( "Enter LONG: " );
            long t  =  in.nextLong ( );
            System.out.println ( "You entered: " + t);
    }
}
```

# PRACTICE

**Program 09_04** Write a program that:

a) asks the user to enter the first name and the last name,
b) stores the first name in *firstName*, a String variable, and the last name in *lastName*, a String variable,
c) concatenates the values of *firstName* and *lastName* and assigns the result to *output*, a String variable,
d) prints the value of *output*.

# PRACTICE - ANSWER

```java
import java.util.*;

public class Prog09_04 {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);

        System.out.print("Enter FIRST NAME: ");
        String firstName = in.nextLine();

        System.out.print("Enter LAST NAME: ");
        String lastName = in.nextLine();

        String output = firstName + " " + lastName;

        System.out.println("Full Name: " + output);
    }
}
```

## *JOptionPane* class

**The *JOptionPane* class:**

- **Contained in the *javax.swing* package.**
- **Contain methods tailored for input / output**

  **- *showInputDialog ( )* : to obtain data from the user**

  **- *showMessageDialog ( )* : to show data in a dialog box**

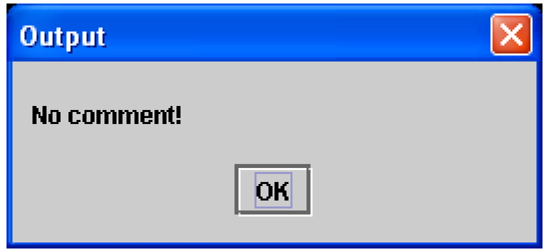
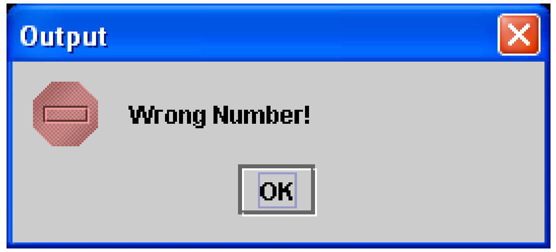
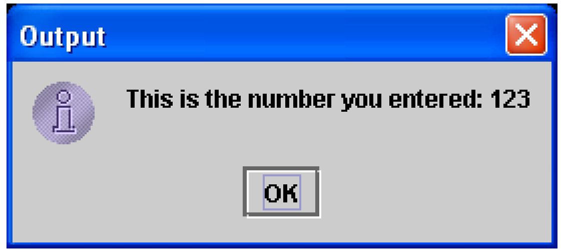
http://java.sun.com/

## *JOptionPane* class: Try it yourself

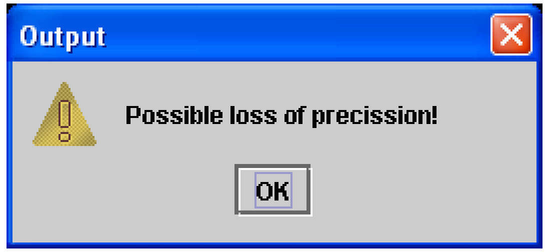
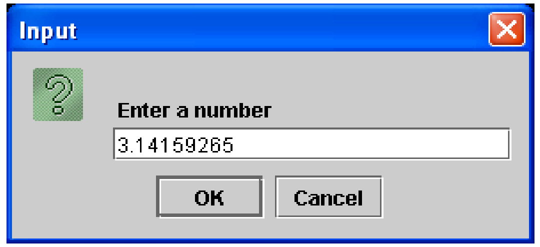
```
// Prog09_05 : The JOptionPane class
import javax.swing.*;
public class Prog09_05
{
    public static void main ( String args[ ] )
    {
            int n;
            String s;
            s = JOptionPane.showInputDialog ( null, "Enter a number",
                                    "Input", JOptionPane.QUESTION_MESSAGE);
            n = Integer.parseInt(s);
            s = "This is the number you entered: " + n + "\n\n";

            JOptionPane.showMessageDialog ( null, s, "Output",
                                    JOptionPane.INFORMATION_MESSAGE);
    }
}
```

# Message Dialog Types

| Message | Symbol | Example |
|---|---|---|
| JOptionPane.PLAIN_MESSAGE (or -1) | No icon |  Output No comment! OK |
| JOptionPane.ERROR_MESSAGE (or 0) | Stop sign |  Output Wrong Number! OK |
| JOptionPane.INFORMATION_MESSAGE (or 1) | "i" |  Output This is the number you entered: 123 OK |

# Message Dialog Types

| Message | Symbol | Example |
|---|---|---|
| JOptionPane.WARNING_MESSAGE (or 2) | "!" |  Output<br>Possible loss of precission!<br>OK |
| JOptionPane.QUESTION_MESSAGE (or 3) | "?" |  Input<br>Enter a number<br>3.14159265<br>OK Cancel |

# The Java Language

## 10. Java Operators

# Operators: precedence of operations

| Priority | Type | Symbol | Associativity |
|---|---|---|---|
| 17 | | ( )  [ ] | Left to right |
| 16 | Unary | var++  var-- | Right to left |
| 15 | Unary | ++var  --var | Left to right |
| 14 | Unary | ~  !  -var  +var | Right to left |
| 15 | Casting | casting | ” |
| 12 | Arithmetic | *  /  % | Left to right |
| 11 | Arithmetic | +  - | ” |
| 10 | Shift | <<  >>  >>> | ” |
| 9 | Relational | instanceof  <  <=  >  >= | ” |
| 8 | Relational | ==  != | ” |
| 7 | Bitwise | & | ” |
| 6 | Bitwise | ^ | ” |
| 5 | Bitwise | \| | ” |
| 4 | Logical | && | ” |
| 3 | Logical | \|\| | ” |
| 2 | Conditional | ?: | Right to left |
| 1 | Assignment | =  *=  /=  %=  +=  -= | Right to left |
| | | <<=  >>=  >>>= | |
| | | &=  ^=  \|= | |

# The Java Language

## 11. The assignment operator

# Operators: *assignment* operator

| Priority | Type | Symbol | Associativity |
|---|---|---|---|
| 17 | | ( ) [ ] | Left to right |
| 16 | Unary | var++ var-- | Right to left |
| 15 | Unary | ++var --var | Left to right |
| 14 | Unary | ~ ! -var +var | Right to left |
| 15 | Casting | casting | ” |
| 12 | Arithmetic | * / % | Left to right |
| 11 | Arithmetic | + - | ” |
| 10 | Shift | << >> >>> | ” |
| 9 | Relational | instanceof < <= > >= | ” |
| 8 | Relational | == != | ” |
| 7 | Bitwise | & | ” |
| 6 | Bitwise | ^ | ” |
| 5 | Bitwise | \| | ” |
| 4 | Logical | && | ” |
| 3 | Logical | \|\| | ” |
| 2 | Conditional | ?: | Right to left |
| 1 | Assignment | = *= /= %= += -= <<= >>= >>>= &= ^= \|= | Right to left |

## Operators: assignment operator

**<u>Examples of use</u>:**

► 
```
num = 23.4546;
 x = y;
 x = y = z = 3;
```

► 
```
double a, b;
a = 2.3;
b = -1.50;
```
=
```
double a = 2.3, b = -1.50;
```

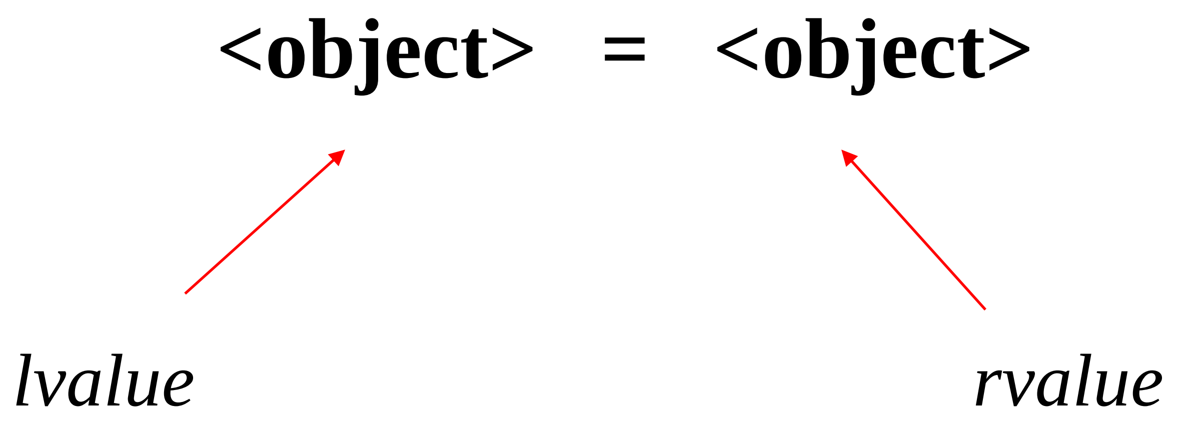
## Operators: assignment operator

**Examples of use (cont.):**

▶ This is **incorrect** (the compiler will report an error)

*a + b = c;*

*(Correct: c = a + b;)*



# The Java Language

## 12. Arithmetic Operations

# Arithmetic Operations

ARITHMETIC OPERATORS:

| | | |
|---|---|---|
| Addition | (+) | |
| Subtraction | (-) | |
| Multiplication | (*) | |
| Division | (/) | |
| Prefix + | (+) | [+<var>] |
| Prefix - | (-) | [–<var>] |
| Modulus | (%) | [*returns remainder*] |

# Operators: *arithmetic* operators

| Priority | Type | Symbol | Associativity |
|---|---|---|---|
| 17 | | ( ) [ ] | Left to right |
| 16 | Unary | var++ var-- | Right to left |
| 15 | Unary | ++var --var | Left to right |
| 14 | Unary | ~ ! -var +var | Right to left |
| 15 | Casting | casting | ” |
| 12 | Arithmetic | * / % | Left to right |
| 11 | Arithmetic | + - | ” |
| 10 | Shift | << >> >>> | ” |
| 9 | Relational | instanceof < <= > >= | ” |
| 8 | Relational | == != | ” |
| 7 | Bitwise | & | ” |
| 6 | Bitwise | ^ | ” |
| 5 | Bitwise | \| | ” |
| 4 | Logical | && | ” |
| 3 | Logical | \|\| | ” |
| 2 | Conditional | ?: | Right to left |
| 1 | Assignment | = *= /= %= += -= <<= >>= >>>= &= ^= \|= | Right to left |

## Operators

<u>Fragment of program</u>:

Addition operator

```
{
                    .
                    .
                    .
              x  =  a  +  b;
                    .
                    .
                    .
}
```

# Operators: adding two numbers

```
import java.util.*;

public class Prog12_01
{
   public static void main(String args[])
   {
      Scanner in = new Scanner(System.in);

      System.out.print("Enter a number: ");
      int x = in.nextInt();

      System.out.print("Enter another number: ");
      int y = in.nextInt();

      System.out.println("The sum is: " + (x+y));
   }
}
```