



# VARIABLES & OPERATORS

**Algorithms and Programming**  
Renny P. Kusumawardani  
Department of Information Systems  
ITS

# BASIC STRUCTURE OF JAVA PROGRAMS

- Programs in the Java programming language consist of a set of classes (classes)
- A class has members, namely fields (variables) and methods (functions)
- To be able to run the program, we need a class that has a main method
- Signature of the function: `public static void main(String[] args)`

# EXAMPLE

```
public class Hello2 {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
        System.out.println();  
        System.out.println("This program produces");  
        System.out.println("four lines of output");  
    }  
}
```

# BASIC CLASS STRUCTURE (WITH MAIN FUNCTION)

```
class <name> {  
    public static void main(String[] <usually named  
    args>) {  
        <statement(s)>;  
    }  
}
```

# JAVA TERMINOLOGY

## **class:**

- (a) A module that can contain executable code.
- (b) A description of a type of objects. (seen later)

**statement:** An executable piece of code that represents a complete command to the computer.

- every basic Java statement ends with a semicolon ;

**method:** A named sequence of statements that can be executed together to perform a particular action or computation.

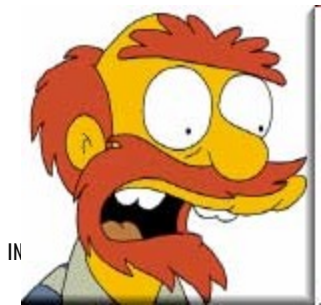
# SYNTAX AND SYNTAX ERRORS

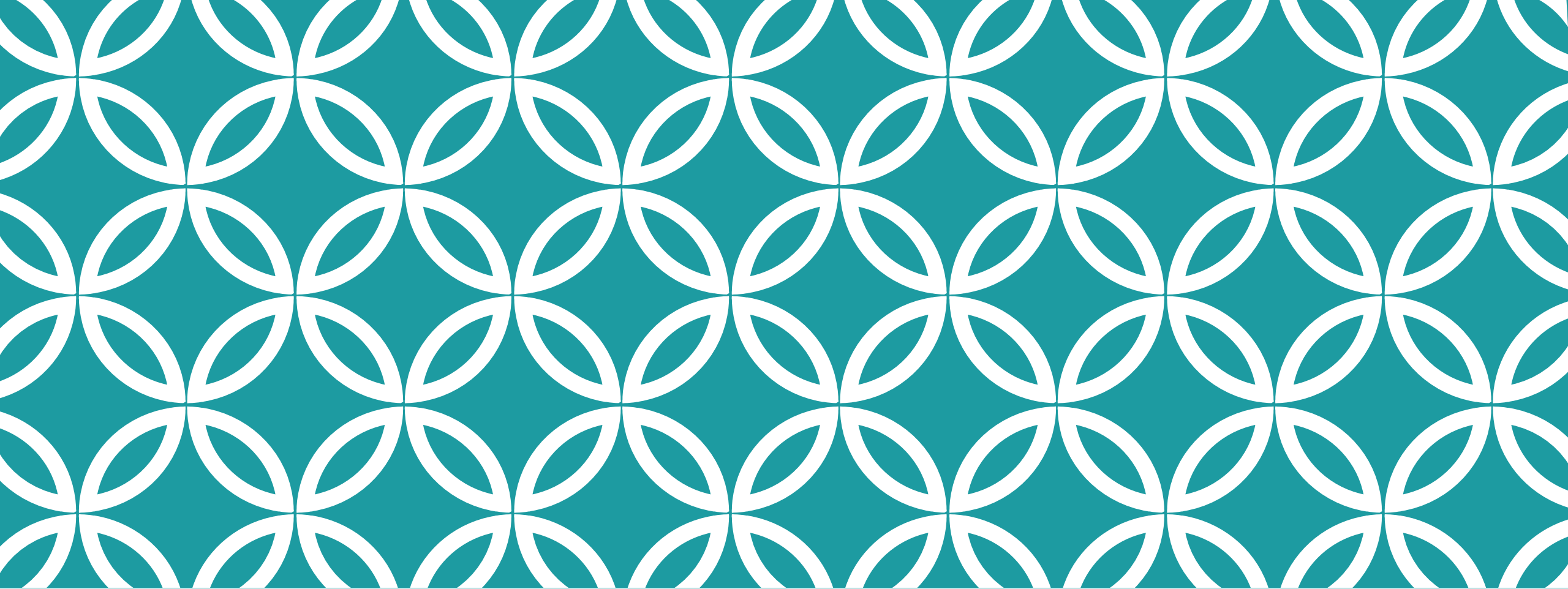
**syntax:** The set of legal structures and commands that can be used in a particular programming language.

**syntax error** or **compiler error:** A problem in the structure of a program that causes the compiler to fail.

- If you type your Java program incorrectly, you may violate Java's syntax and see a syntax error.

```
public class Hello {  
    pooblic static void main(String[] args) {  
        System.owt.println("Hello, world!")_  
    }  
}
```





# VARIABLES AND DATA TYPES

# VARIABLES

The function of a variable in programming is similar to the function of a variable in mathematics

- As a 'container' to accommodate a value that can change
- Has a name, by which we can refer to the value stored in the variable

In the Java programming language, at the time of declaration (creation, birth) of a variable, in addition to its name, we also need to specify the type of data that can be stored in that variable.



# DATA TYPES

Why do we need to declare the data type of a variable?

1. To tell how the variable needs to be stored
  - What is the allocation of the required storage size in memory
  - Where the variable needs to be stored (on the stack or heap – will learn later)
2. To tell what operations can be performed on the variable. The interpretation of an operator is also sometimes affected by the type of the variable

# DATA TYPE EXAMPLE

**boolean**: stores truth value (true or false)

**int**: stores integers, e.g. 0, 1, -47

- Stores integers from  $-2^{31}$  to  $2^{31} - 1$
- int-like data type, but with a wider range of possible values →  
long

**double**: stores real values, e.g. 3.14, 1.0, -2.1

- The default data type for decimal values (double precision)
- Like double, but with less precision → float (single precision)
- NOTE! The convention for decimal numbers in Java by default follows the convention for writing numbers in English, namely the decimal point uses a dot '.' (not a comma ',' as in Indonesian)

**String**: stores texts, for example "hello", "example"

For more, you could take a look at (don't worry if you don't yet understand all of it):

<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

# VARIABLE DECLARATION

Syntax:

**Data\_type Variable\_name;**

Example:

**String foo;**

**int x;**

**long aLongNumber;**

**boolean isStudent;**

# OPERATORS

Operator is a symbol that is used to perform operations on variables (similar to operators in math)

Some basic operators:

Assignment: =

Addition: +

Subtraction: -

Multiplication: \*

Division: /

# TIPE-TIPE OPERATOR

## 1. Aritmatika

+ (tambah), - (kurang), \* (kali), / (bagi), % (modulo)

## 2. Logika

& (and), | (or), && (and), || (or), == (equal), != (not equal), ! (negasi)

## 3. Manipulasi Bit

>>, <<

## 4. Assignment operator

=

# OPERATOR PRECEDENCE

Order of precedence: if a statement line contains several operators, which operation will be performed first?

For the basic operators above, the order is the same as in the math:

Priority 1: opening parenthesis – closing parentheses (parentheses)

Priority 2: multiplication and division

Priority 3: addition and subtraction

If there are several operators with the same precedence priority, then the operation is performed from left to right.

# DATA TYPE DEPENDENT OPERATOR INTERPRETATION: + ON STRING OPERANDS

Operands are data that are operated by operators

Example: in the statement `1 + 2`, operator `+` operates on operands 1 and 2

In the String operand, `+` acts as a concatenator: concatenates two Strings into a new String

- Example: `"I" + "eat"` will result in `"Ieat"`
- If one of the operands in the `+` operator is a String, a new String will be generated with the other operand being interpreted as a String
- Example: `"the number of my siblings is: " + 3` will produce `"the number of my siblings is: 3"`



# VARIABLE VALUE ASSIGNMENT

Giving value to a variable is called an assignment and is performed using the operator '='

**Example:**

```
String foo;
```

```
foo = "I loooveeeee programming!!!";
```

foo will contain the String: "I loooveeeee programming!!!"

(You could display the content of foo using: `System.out.println(foo);`)

Variable declaration can be done together with assignment, for example:

```
int x = 1;
```

```
String foo = "I loooveeeee programming!!!"
```

# NOW, LET'S PUT THESE TOGETHER...

```
class DoMath {  
    public static void main(String[] arguments) {  
        float score = (float) (1.0 + 2.0 * 3.0);  
        System.out.println(score);  
        score = score / 2.0;  
        System.out.println(score);  
    }  
}
```

# ANOTHER EXAMPLE

```
class DoMath2 {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        double copy = score;  
        copy = copy / 2.0;  
        System.out.println(copy);  
        System.out.println(score);  
    }  
}
```

# EXERCISE: FROM FORMULA TO PROGRAM

Write a program to calculate the position of an object falling at a time using the following formula:

$$x(t) = 0.5 \times at^2 + v_i t + x_i$$

# ... HERE'S AN EXAMPLE ANSWER

```
class GravityCalculator {  
    public static void main(String[] args) {  
        double gravity = -9.81;  
        double initialVelocity = 0.0;  
        double fallingTime = 10.0;  
        double initialPosition = 0.0;  
        double finalPosition = .5 * gravity * fallingTime *  
                                fallingTime;  
        finalPosition = finalPosition +  
                        initialVelocity * fallingTime;  
        finalPosition = finalPosition + initialPosition;  
        System.out.println("An object's position after " +  
                            fallingTime + " seconds is " +  
                            finalPosition + " m.");  
    }  
}
```