

## PROGRAMMING LANGUAGE

A SET OF RULES,  
SYMBOLS, AND  
SPECIAL WORDS  
USED TO  
CONSTRUCT  
PROGRAMS

## SYNTAX RULES

RULES THAT TELL  
YOU WHICH  
STATEMENTS  
(INSTRUCTIONS)  
ARE LEGAL, OR  
ACCEPTED BY THE  
PROGRAMMING  
LANGUAGE.

## SEMANTIC RULES

RULES THAT  
DETERMINE THE  
MEANING OF THE  
INSTRUCTIONS

COMMENTS  
ARE FOR THE  
READER,  
NOT THE  
COMPILER.

## COMMENTS

## COMMENTS

### 1. SINGLE-LINE COMMENTS

- BEGIN WITH // AND CAN BE PLACED ANYWHERE IN THE LINE
- EVERYTHING ENCOUNTERED AFTER // IS IGNORED BY THE COMPILER

### 2. MULTIPLE-LINE COMMENTS

- ENCLOSED BETWEEN /\* AND \*/
- COMPILER IGNORES EVERYTHING BETWEEN /\* AND \*/

## COMMENTS

```

1 // *****
2 // This is a simple Java program.
3 // It displays "Hello, world!" in
4 // the terminal window.
5 // *****
6
7 public class HelloWorld {
8
9     public static void main(String[] args) {
10         // Prints "Hello, World" to the terminal window.
11         System.out.println("Hello, World");
12     }
13
14 }

```

## COMMENTS



### Good Programming Practice 2.1

Some organizations require that every program begin with a comment that states the purpose of the program and the author, date and time when the program was last modified.

© COPYRIGHT 1992-2018 BY PEARSON EDUCATION, INC. ALL RIGHTS RESERVED.

## COMMENTS

LABS WILL ALL NEED TO BEGIN WITH THE FOLLOWING INFO IN A COMMENT

```

*****
YOUR NAME
CSCI 206 – YOUR SECTION #
LAB ASSIGNMENT NAME
DATE
*****

```



## Good Programming Practice 2.2

Use white space to enhance program readability.

© COPYRIGHT 1992-2018 BY PEARSON EDUCATION, INC. ALL RIGHTS RESERVED.

### *Using Blank Lines*

- Blank lines, space characters and tabs
  - Make programs easier to read.
  - Together, they're known as white space (or whitespace).
  - White space is ignored by the compiler.

© COPYRIGHT 1992-2018 BY PEARSON EDUCATION, INC. ALL RIGHTS RESERVED.

```

1 // Fig. 2.1: Welcome1.java
2 // Text-printing program.
3
4 public class Welcome1 {
5     // main method begins execution of Java application
6     public static void main(String[] args) {
7         System.out.println("Welcome to Java Programming!");
8     } // end method main
9 } // end class Welcome1

```

Welcome to Java Programming!

Fig. 2.1 | Text-printing program.

© COPYRIGHT 1992-2018 BY PEARSON EDUCATION, INC. ALL RIGHTS RESERVED.

## SPECIAL SYMBOLS

+ - \* /

. ; ? ,

<= != == >=

## RESERVED WORDS (KEYWORDS)

- KEYWORDS ARE ALWAYS LOWERCASE
- EACH RESERVED WORD IS CONSIDERED A SINGLE SYMBOL

## IDENTIFIER

- NAMES OF THINGS SUCH AS VARIABLES, CONSTANTS, AND METHODS
- SOME ARE PREDEFINED; SOME ARE USER DEFINED

## JAVA IDENTIFIER

Must consist of:  
Letters  
Digits  
Underscore(\_)   
Dollar sign \$

Must begin with:  
Letter  
Underscore  
Dollar sign

- WHICH OF THE FOLLOWING ARE LEGAL IN JAVA?

## JAVA IDENTIFIER

1. First
2. 2<sup>nd</sup>
3. payrate
4. Counter1
5. \$Amount
6. one+two
7. studentName

SET OF  
VALUES  
TOGETHER  
WITH A SET  
OF  
OPERATIONS  
ON THOSE  
VALUES

## DATA TYPES

- FUNDAMENTAL  
DATA TYPES IN JAVA
- 3 CATEGORIES

## PRIMITIVE DATA TYPES

- FUNDAMENTAL  
DATA TYPES IN JAVA
- 3 CATEGORIES
  - INTEGRAL
- FLOATING-POINT
- BOOLEAN

## PRIMITIVE DATA TYPES

- DEALS WITH  
INTEGERS,  
NUMBERS  
WITHOUT A  
DECIMAL AND  
CHARACTERS

## INTEGRAL

<ul style="list-style-type: none"> <li>• float</li> <li>• DEALS WITH DECIMAL NUMBERS</li> </ul>	<h2>FLOATING-POINT</h2>
---	-------------------------

<ul style="list-style-type: none"> <li>• float</li> <li>• DEALS WITH DECIMAL NUMBERS</li> <li>• TEND TO BE LARGER NUMBERS</li> <li>• 4 BYTES</li> </ul>	<h2>FLOATING-POINT</h2>
---	-------------------------

<ul style="list-style-type: none"> <li>• float</li> <li>• DEALS WITH DECIMAL NUMBERS</li> <li>• TEND TO BE LARGER NUMBERS</li> <li>• 4 BYTES</li> <li>• double</li> <li>• LARGER NUMBERS</li> <li>• 8 BYTES</li> </ul>	<h2>FLOATING-POINT</h2> <h2>DOUBLE</h2>
--	---

<ul style="list-style-type: none"> <li>• float</li> <li>• DEALS WITH DECIMAL NUMBERS</li> <li>• TEND TO BE LARGER NUMBERS</li> <li>• 4 BYTES</li> <li>• double</li> <li>• LARGER NUMBERS</li> <li>• 8 BYTES</li> </ul>	<h2>FLOATING-POINT</h2> <ul style="list-style-type: none"> <li>• Max number of decimal places: 6 or 7</li> </ul> <h2>DOUBLE</h2> <ul style="list-style-type: none"> <li>• Max number of decimal places: typically 15</li> </ul>
--	---

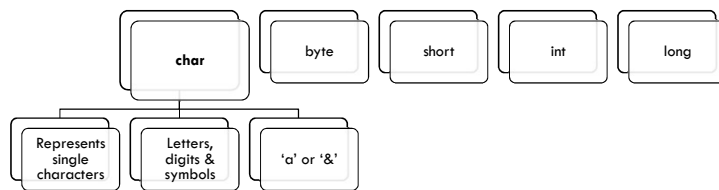
- LOGICAL VALUES
- TRUE OR FALSE

## BOOLEAN

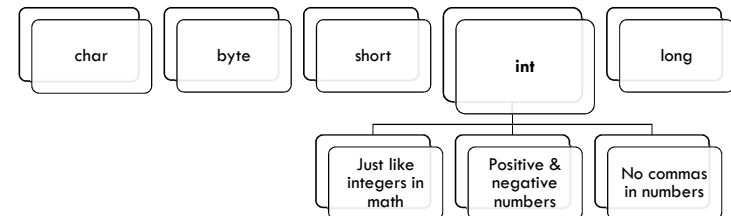
### INTEGRAL DATATYPES



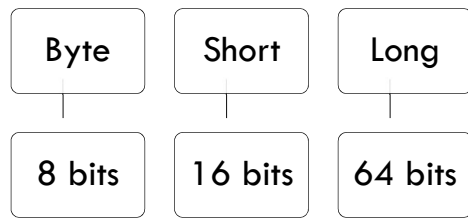
### INTEGRAL DATATYPES



### INTEGRAL DATATYPES



## INTEGRAL DATATYPES



• WHAT DATATYPES WOULD YOU USE?

PRIMITIVE  
DATATYPES

1. 32412
2. @
3. 3.14159265359
4. True
5. 431.7