* *Why java?*
  + Java’s role: Run in a JVM
  + C-like language
  + Compiled language: compiles to a bytecode
  + OOP language
* *Syntax and Semantics*
  + Syntax: structure of symbols in programming language
  + Token: single atomic component of the language (variable, keyword, operator)
  + Semantics: meaning of code
  + Syntax and Semantics:
    - {}: block of code
    - ;:newline kind of thing
    - Java variables have type (different from python)
* *Hello World*

*public class HelloWorld {*

*public static void main(String[] args) {*

*system.out.println(“Hello, World!”);*

*}*

*}*

* *A program development cycle*
  + When write java program: write the code 🡪 compile the code
  + When compile, can run to test it
    - Crash while running = run time error (Debug 🡪 Rewrite 🡪 return compiling step)
    - No error but not run correctly = logical error
* *Java ☺*
  + Classes = definition of object (OOP language)
  + Method = definition to make change to object
  + Statements:
  + Main method = first method on the program
* *Comments in Java:*
  + // goes until the end of the line
  + /\* goes until the matching end is found \*/
  + /\* /\* abc \*/ \*/ = invalid
* *Identifiers:*
  + Variables, methods, classes…
  + Case sensitive
  + Valid names have:
    - Letters, digits, underscore, dollar sign
    - Can’t start with a digit
    - Cannot use a reserved keyword (if, for, class)
  + Common style guidelines:
    - Lower case for variables/methods: count, numDays
    - Upper case for class name
* *Data types:*
  + Primitive types: numbers, booleans, characters
  + Reference types: class, strings
  + Booleans: true/false
  + Characters: char, single quoted
    - \n: newline
  + Numbers: integer, floating
* Integer types:
  + Decimal: without reading zero (except number zero itself)
  + Octal (base 8): number with leading zero (012)
  + Binary (base 2): number begin with 0b or 0B (0b10110)
  + Hexadecimal (base 16): numbers beginning with 0x (0x3, 0xA4)
  + Long data value: number with L or 1 after (12225L)