Code Editors

Windows: Notepad++
Sublime Text

Mac: Text Wrangler
Linux: Sublime Text

- comments
- class HelloWorld
- main() heading
- System.out.println()}
Compiling & Running:

HelloWorld.java  →  Source file
                 ↓
Java Compiler   →  iawac
                 ↓
HelloWorld.class →  Executable file
                 ↓
Java Virtual Machine  →  java
                 ↓
Program Output

Read chapter 4 at Text.
Chapter 2: The Java Language

- White space: space, tab, newline
  - Free format: extra whitespace is ignored.

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

are equivalent.
acceptable styles:

```java
class HelloWorld {
    indent 3-5 spaces (not tab)
}
```

or

```java
class HelloWorld {
    }
}
```

Do not use tabs anywhere!
// single line comment

/*
   multiple line comments
*/

/****
   multi-line comment used by javadoc
*/
Reserved Words

Java has 58 keywords

so far: class, public, static, void

note: main
Hello World
args
System.out.println

are not keywords.
- **Identifiers**
  - must follow rules
    - must consist only of
      - letters: a, ... z, A, ... Z
      - underscores: _
      - digits: 0, 1, 2, ..., 9
      - dollar sign: $ (don't use!)
    - may not begin with a digit
    - may not be a reserved word
      - may not be: true, false, null.
Ex.

- happy
- unhappy
- unhappy

HAPPY

class

2 class

class 2

class

a_long_variable_name

aLongVariableName

"a long variable name"
Data Types

defined by the way data is encoded in binary.

Two categories of data types

- Primitive types (8):
  byte, short, int, long
  float, double, char, boolean

- Reference types
  Invented by programmers
  String, .... others
  by convention begin with capital.
<table>
<thead>
<tr>
<th>Type</th>
<th># bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>8</td>
<td>Integer</td>
</tr>
<tr>
<td>short</td>
<td>16</td>
<td>&quot;</td>
</tr>
<tr>
<td>int</td>
<td>32</td>
<td>&quot;</td>
</tr>
<tr>
<td>long</td>
<td>64</td>
<td>&quot;</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
<td>Real number</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>&quot;</td>
</tr>
<tr>
<td>char</td>
<td>16</td>
<td>Character</td>
</tr>
<tr>
<td>boolean</td>
<td>≥1</td>
<td>false, true</td>
</tr>
</tbody>
</table>
Literals are program representation of values in these types:

<table>
<thead>
<tr>
<th>Type</th>
<th>Literal example</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>1, -57, 3954</td>
</tr>
<tr>
<td>$n$</td>
<td># strings</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
<td>0, 1</td>
</tr>
<tr>
<td>2</td>
<td>00, 01, 10, 11</td>
</tr>
<tr>
<td>3</td>
<td>000, 001, 010, 011, 100, 101, 110, 111</td>
</tr>
<tr>
<td>4</td>
<td>0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111</td>
</tr>
</tbody>
</table>

$\#$ bit strings of length $n) = 2^n$