Structure of Java

zyBooks Chap 1.5, 1.6, 2.1, 2.2

Structure of Java

- Class
  - Method
    - Statement
- Documentation/Comments
- Spacing

```java
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Documentation/Comments

- A **note** for programmers that **describes** or **clarifies** the code
  - **Not readable** to computers
  - **NOT executed** when the program runs

Javadoc → /.....*/**

```
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Single-line Comments → // .....

Read more in Programming Style Guide (Brightspace | Content | Course Documents)

Class – A program

- Class Header
  - Capitalize the first letter of each word, no space in between (e.g. HelloWorld)

- The **file name** (HelloWorld.java) must **match exactly** with the **class name**, including capitalization
  - Java is **case-sensitive**!!!

```java
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Matching braces { ... }

Method – A named group of statements

- Method Header
  - Begin with a lowercase letter, capitalize the first letter of the attached words

- Every **executable** Java program consists of a class, that **contains** a method named **main** that contains the statements to be executed

```java
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Matching braces { ... }

Statement – An instruction to be executed

- **Ends with semi-colon** (;)

```java
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Print/Println Statement

- System.out is an **object** for sending output to the screen
- println is a **method** to print whatever is **inside parentheses** to the screen, in this case, a String "Hello World!"
  - The item(s) inside parentheses are called **parameter(s)** or **argument(s)**

```java
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Spacing

- The best way to make your code readable is to **indent** nested code
- Indent every time you **go inside braces**
- You must indent using **four spaces** or **one tab** (manually set it to four spaces)

```
/**
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Read more in Programming Style Guide (Brightspace | Content | Course Documents)

Identifier

- Identifier is a **name** given to an entity in a program, such as a class name
- Identifiers **start with a letter** and are followed by a number of letters or digits. Letters include:
  - Alphabetic characters, upper and lower case (A-Z, a-z)
  - Underscore (_)
  - Dollar sign ($)
- Identifiers should be descriptive/meaningful

Q: Indicate if each of the following Java identifier is legal or not. If not, why?

- 3Example ✗ Starts with a digit
- varTest ✓
- max_value ✓
- max-value ✗ Contains -
- quiz+HW ✗ Contains +
- quiz 1 ✗ Contains space
- system ✓

Java Keywords

An identifier that you cannot use because it already has a **reserved** meaning in Java.

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	