Structure of Java

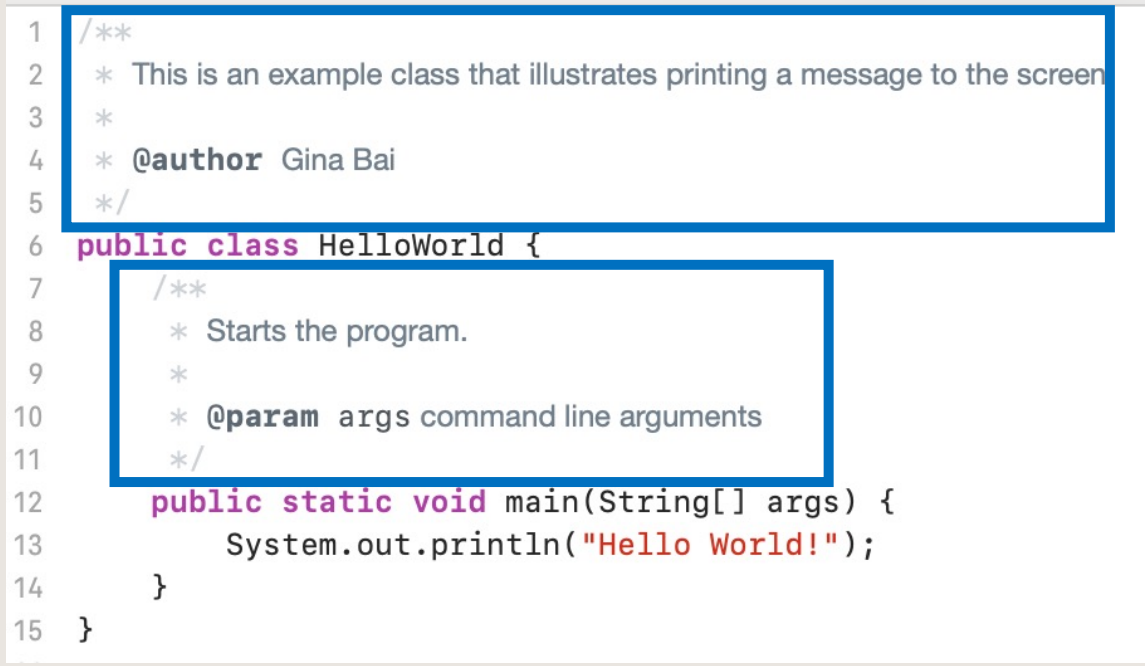
Structure of Java

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

```java
/**
 * This is an example class that illustrates printing a message to the screen
 *
 * @author Gina Bai
 */
public class HelloWorld {
    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Documentation/Comments

- Comment
  - A note written in source code by the programmer to describe or clarify the code
- Comments are **not executed** when your program runs.
- See Programming Style Guide (Brightspace > Content > Course Documents)



```
1  /**
2   * This is an example class that illustrates printing a message to the screen
3   *
4   * @author Gina Bai
5   */
6  public class HelloWorld {
7      /**
8       * Starts the program.
9       *
10      * @param args command line arguments
11      */
12     public static void main(String[] args) {
13         System.out.println("Hello World!");
14     }
15 }
```

Class – A program

- Class Header
  - Naming convention: capitalize the first letter of each word, no space in between (e.g. HelloWorld)
- The filename (HelloWorld.java) must **match exactly** with class name, including capitalization
  - Java is "**case-sensitive**"
- Matching braces {} (lines 6 & 15)

```
1  /**
2   * This is an example class that illustrates printing a message to the screen
3   *
4   * @author Gina Bai
5   */
6  public class HelloWorld {
7      /**
8       * Starts the program.
9       *
10      * @param args command line arguments
11      */
12     public static void main(String[] args) {
13         System.out.println("Hello World!");
14     }
15 }
```

Method – A named group of statements

- Method Header
  - Naming convention: 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.
- Matching braces {} (lines 12 & 14)

```
1  /**
2   * This is an example class that illustrates printing a message to the screen
3   *
4   * @author Gina Bai
5   */
6  public class HelloWorld {
7      /**
8       * Starts the program.
9       *
10      * @param args command line arguments
11      */
12     public static void main(String[] args) {
13         System.out.println("Hello World!");
14     }
15 }
```

Statement – An instruction to be executed

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

```
1  /**
2   * This is an example class that illustrates printing a message to the screen
3   *
4   * @author Gina Bai
5   */
6  public class HelloWorld {
7      /**
8       * Starts the program.
9       *
10      * @param args command line arguments
11      */
12     public static void main(String[] args) {
13         System.out.println("Hello World!");
14     }
15 }
```

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
- See Programming Style Guide (Brightspace > Content > Course Documents)

```java
/**
 * This is an example class that illustrates printing a message to the screen
 *
 * @author Gina Bai
 */
public class HelloWorld {
    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Identifier

- Identifier is a **name** given to an entity in a program, such as 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

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	