

## 2. First Program in Java

Java

**Fall 2009**

*Instructor: Dr. Masoud Yaghini*

## Outline

---

- HelloWorld Program
- References



# HelloWorld Program

# The "Hello World!" Program

- Your first program, [HelloWorld](#), will simply display the greeting "Hello World!".
- To write your first program, you'll need:
  - The Java SE Development Kit 6 (JDK 6)
  - An IDE (**NetBeans** IDE 6.0)
- [HelloWorld](#) program:
  - [HelloWorld.java](#)

# The "Hello World!" Application

- The "Hello World!" application consists of three primary components:
  - Source code comments
  - The `HelloWorld` class definition
  - The `main` method

# First Program in Java

## The Comments

- The following text defines the comments:

```
/**  
 * The HelloWorldApp class implements an application that  
 * simply prints "Hello World!" to standard output.  
 */
```

- Comments are ignored by the compiler but are useful to other programmers.

# Source code comments

- The Java programming language supports three kinds of comments:
  - `/* text */`
    - The compiler ignores everything from `/*` to `*/`.
  - `/** documentation */`
    - This indicates a documentation comment (doc comment, for short).
    - The compiler ignores this kind of comment, just like it ignores comments that use `/*` and `*/`.
    - The **javadoc tool** uses doc comments when preparing automatically generated documentation.
  - `// text`
    - The compiler ignores everything from `//` to the end of the line.

# HelloWorld class definition

- HelloWorld class definition:

```
public class name  
{  
  ...  
}
```

- The keyword `class` begins the class definition for a **class named** name, and the code for each class appears between the opening and closing curly braces marked in bold above.
- We will classes in detail later.
- For now it is enough to know that every application begins with a class definition.



# The main Method

---

- The following text begins the definition of the main method:

```
public static void main(String[] args)
```

# The *main* Method

- In the Java programming language, every application must contain a `main` method whose signature is:

```
public static void main(String[] args)
```

- It's the entry point for your application and will subsequently call up all the other methods required by your program.
- The modifiers `public` and `static` can be written in either order (`public static` or `static public`) but the convention is to use `public static`

### The *main* Method

- The `main` method accepts a single argument: an array of elements of type `String`.

```
public static void main(String[] args)
```

- You can name the argument anything you want, but most programmers choose "`args`" or "`argv`."
- This array is the mechanism through which the runtime system passes information to your application.

### The *main* Method

- Each string in the array is called a **command-line argument**.
- **Command-line arguments** let users affect the operation of the application without recompiling it.
- The "**Hello World!**" application ignores its command-line arguments, but you should be aware of the fact that such arguments do exist.

### The *main* Method

- Finally, the following line uses the System class from the API to print the "Hello World!" message to standard output:

```
System.out.println("Hello World!");
```



# References



### References

---

- S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, **The Java Tutorial: A Short Course on the Basics**, 4th Edition, Prentice Hall, 2006. (Chapter 1)

***The End***

