LECTURE 1

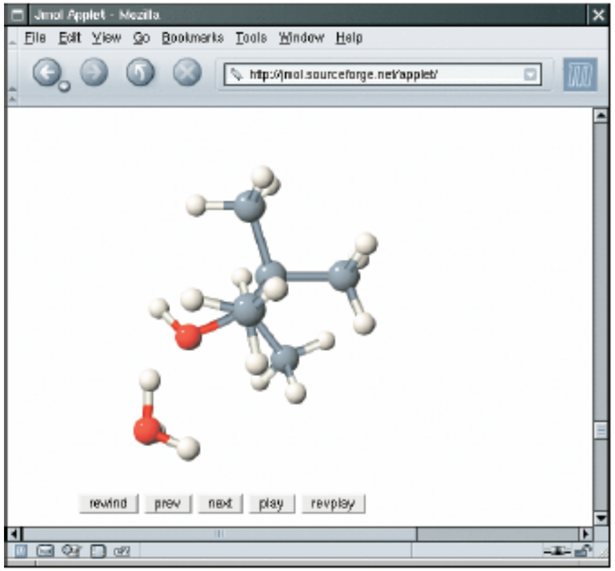
INTRODUCTION TO JAVA

High-level Comparison

Python	Java
Concise	Verbose
Indentation	Braces { }
Interpreted	Compiled
Statically typed	Dynamically typed
Used in data science, everywhere	Used in enterprise software, android
Cross-platform, object-oriented, readable	Cross-platform, object-oriented, readable

1.3 The Java Language

- In 1991, James Gosling of Sun Microsystems designed what would become the Java programming language
- Java was originally designed for programming consumer devices, but it was first successfully used to write Internet applets
  - An applet is typically embedded inside a web page and runs in the context of a browser

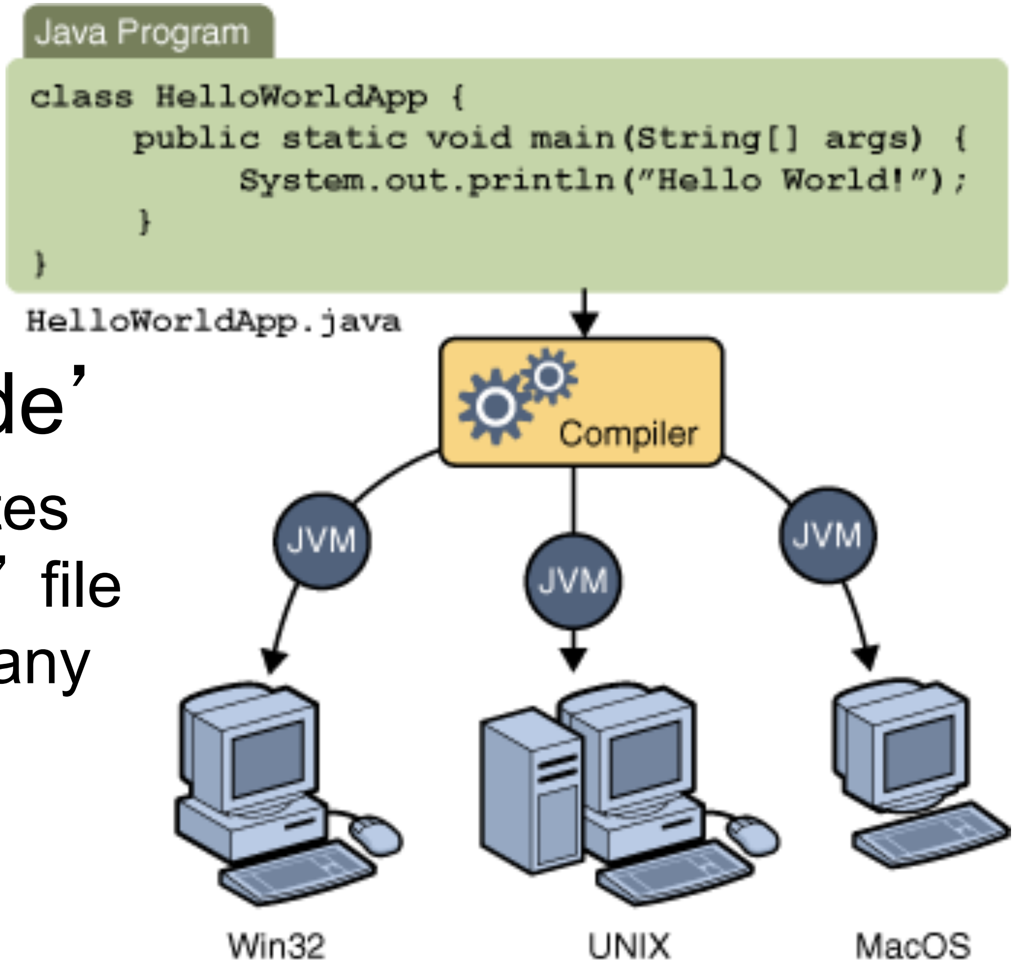


Java History

- Java Design Goals
  - Safe: Can be run inside a browser and will not attack your computer
  - Portable: Run on many Operating Systems
    - Windows
    - Mac OS
- Java programs are distributed as instructions for a ‘virtual machine,’ making them platform-independent
  - Virtual machines are available for most Operating Systems. The iPhone is a notable exception

Java Virtual Machines

- Source code
- Portable ‘byte code’
  - The compiler generates byte code in a ‘class’ file which can be run on any Java Virtual Machine
- Oracle Tutorials



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

Python vs. Java

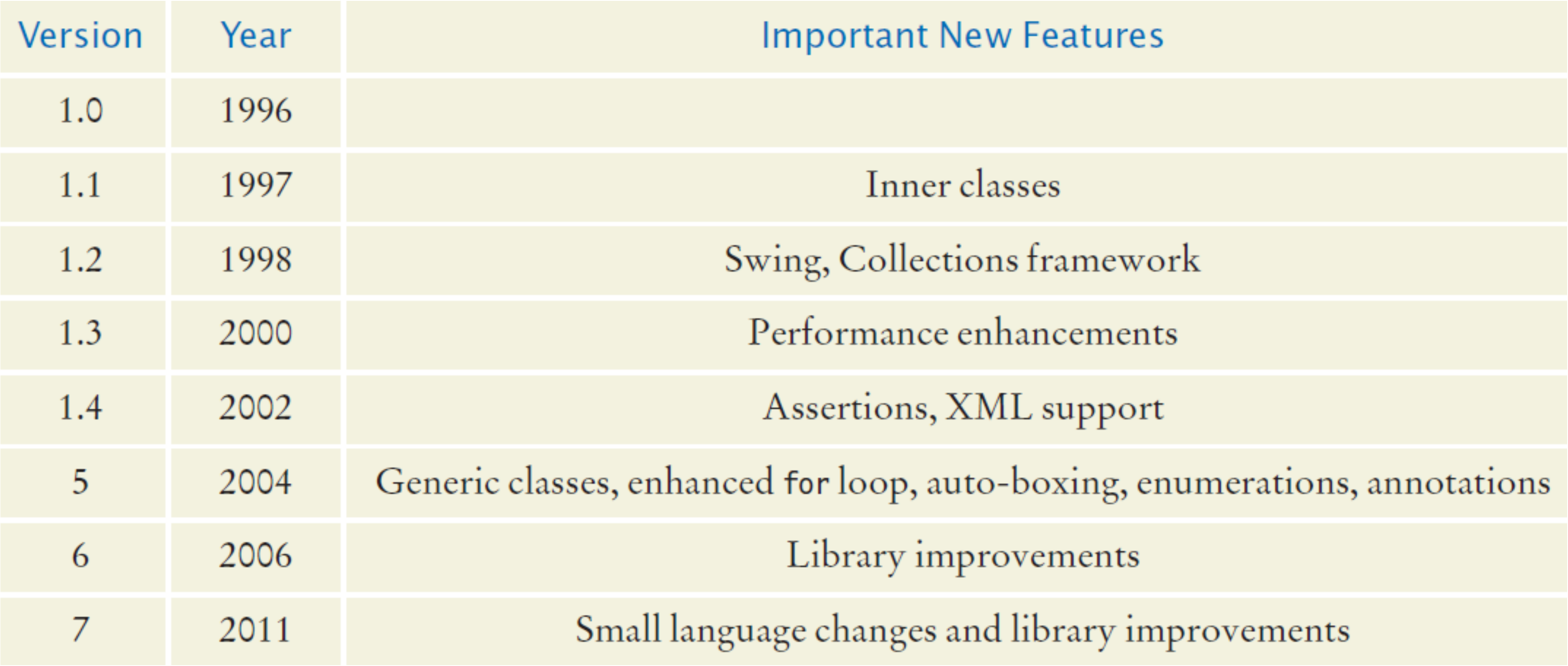
Interpreter

- Looks one line at a time
- Memory efficient
- Faster to prototype
- Errors found at runtime

Compiler

- Looks at entire file at a time
- Execution efficient
- Errors found at compile time
- Other errors at runtime

Java Timeline

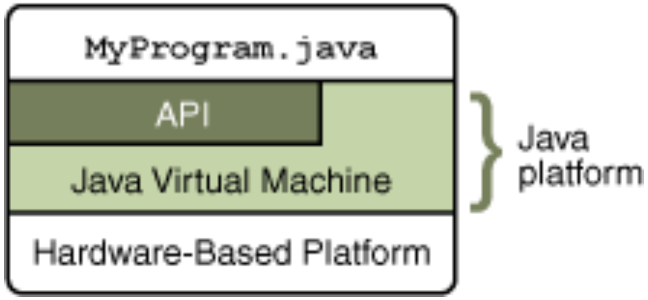


Version	Year	Important New Features
1.0	1996	
1.1	1997	Inner classes
1.2	1998	Swing, Collections framework
1.3	2000	Performance enhancements
1.4	2002	Assertions, XML support
5	2004	Generic classes, enhanced for loop, auto-boxing, enumerations, annotations
6	2006	Library improvements
7	2011	Small language changes and library improvements

- Version 8 released in 2014
- http://oracle.com.edgesuite.net/timeline/java/

The Java API

- The Java Platform consists of two parts:
  1) Java Virtual Machine
  2) Java API



- The Application Programming Interface (API) is a large collection of software packages that programmers can use:
  - Graphics, user interface, networking, sound, database, math, and many more

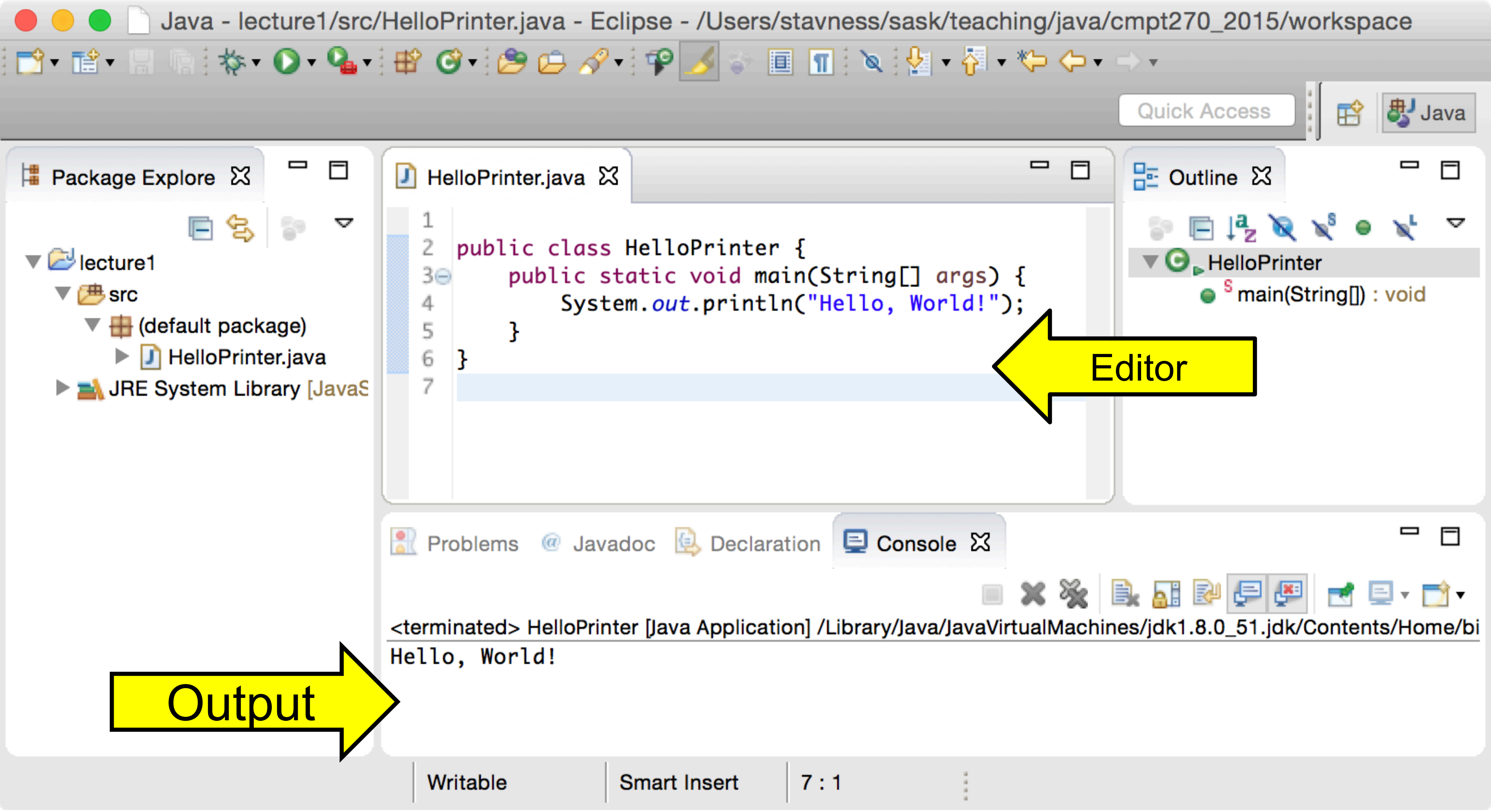
The Java SDK

- You need to install the Java Software Development Kit (JDK) to create Java programs
  - www.oracle.com
  - Download > **Java SE Development Kit 8u60**
  - Location after installed:
    - `C:\Program Files\Java\jdk1.8.x`
    - `/Library/Java/JavaVirtualMachines/jdk1.8.x`
- The JDK includes programs such as:
  - `java` (Executes Java applications)
  - `javac` (Java compiler)
  - `javadoc` (Javadoc generator)

1.4 Programming Environment

- There are many free programming tools available for Java
- Components of an Integrated Development Environment (IDE):
  - Source code editor helps programming by:
    - Listing line numbers of code
    - Color lines of code (comments, text…)
    - Auto-indent source code
  - Output window
  - Debugger
- We will use Eclipse — www.eclipse.org

Eclipse



Java - lecture1/src/HelloPrinter.java - Eclipse - /Users/stavness/sask/teaching/java/cmpt270_2015/workspace

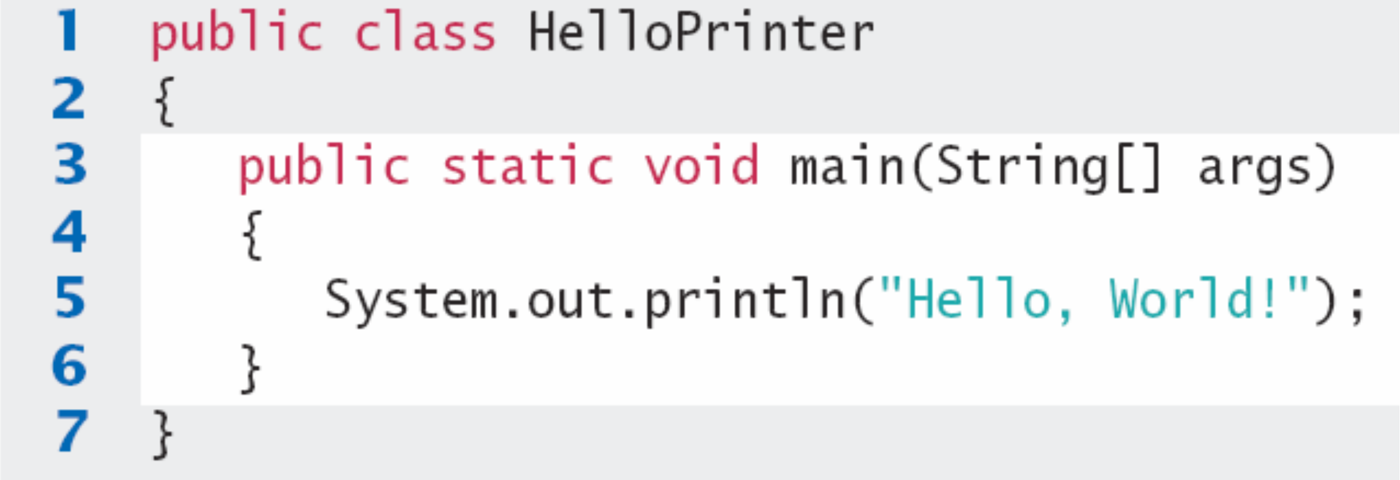
```java

public class HelloPrinter {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}

```

Editor

```
<terminated> HelloPrinter [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/bi
Hello, World!
```

Output

Hello world program in Java



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

- We will examine this program in the next section
  - Typing it into your IDE would be good practice!
  - Be careful of spelling
  - JaVa iS CaSe SeNsItiVe
  - Java uses special characters, e.g. `{ } ( ) ;`

Entry points

Python

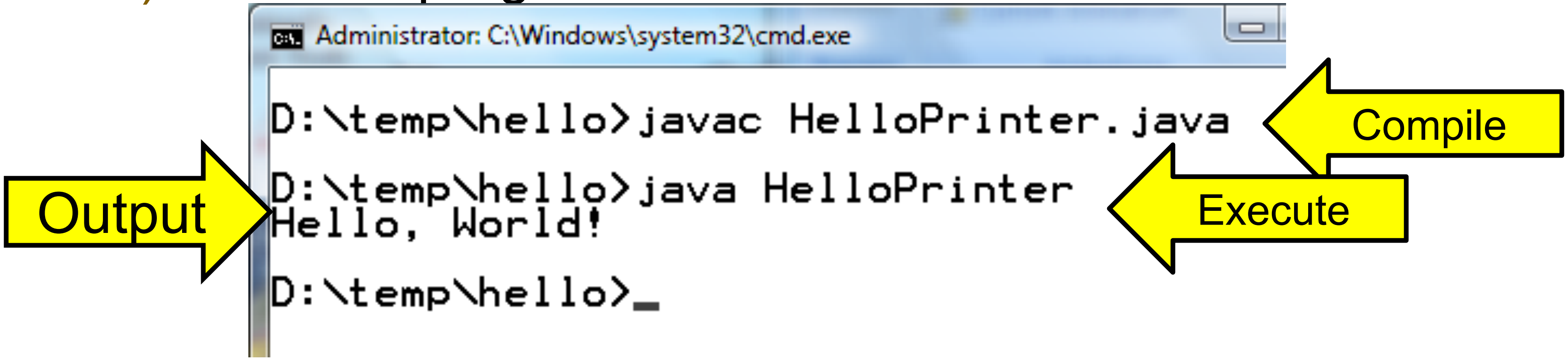
```python
def main():
   print("Hello World!")

if __name__ == '__main__':
    main()
```

Java

```java
public class HelloWorld {

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

Text Editor Programming

- You can also use a simple text editor such as Notepad, Sublime, VI, Emacs, etc. to write your source code
- Once saved as `HelloPrinter.java`, you can use a console window to:
  1) Compile the program
  2) Run the program



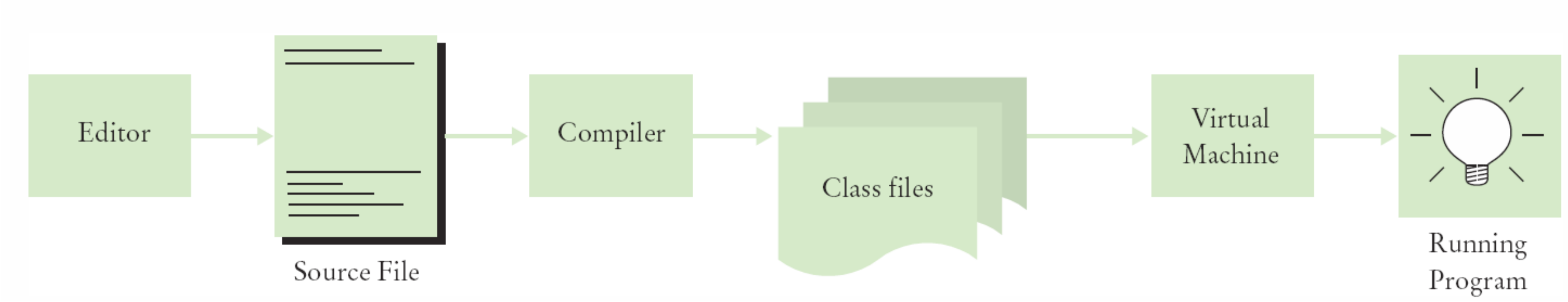
```
D:\temp\hello>javac HelloPrinter.java

D:\temp\hello>java HelloPrinter
Hello, World!

D:\temp\hello>_
```

Compile

Output

Execute

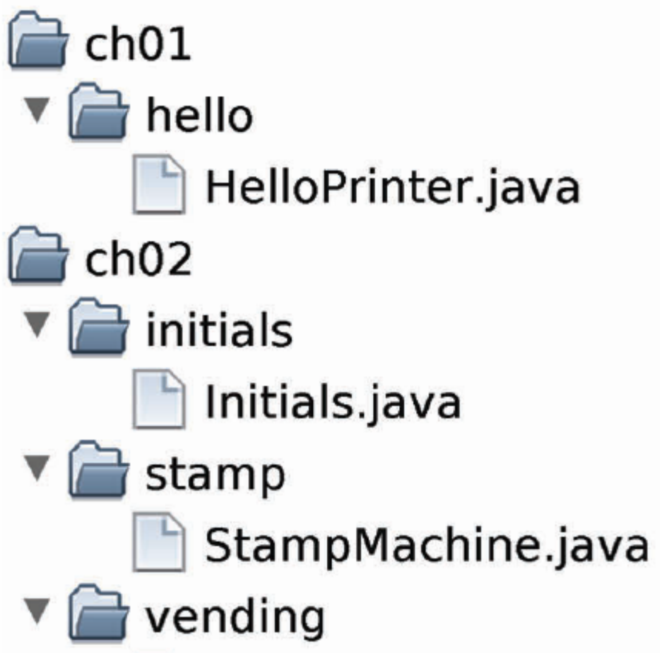
Source Code to Running Program



- The compiler generates the `.class` file which contains instructions for the Java Virtual machine
- Class files contain ‘byte code’ that you cannot edit
  - `D:\temp\hello>Type HelloPrinter.class`
  - `╩■║╛    2 ↔ ♠ ☼        ► ↕ ‼ ¶ § ▬☺ ♠<init>☺ ♥()V☺ ♦Code☺ ☼LineNumberTable☺ ♦main▬([Ljava/lang/String;)V☺`
  - `Hello, World! elloPrinter.java♀ ↨♀ ↑ ↓☺`

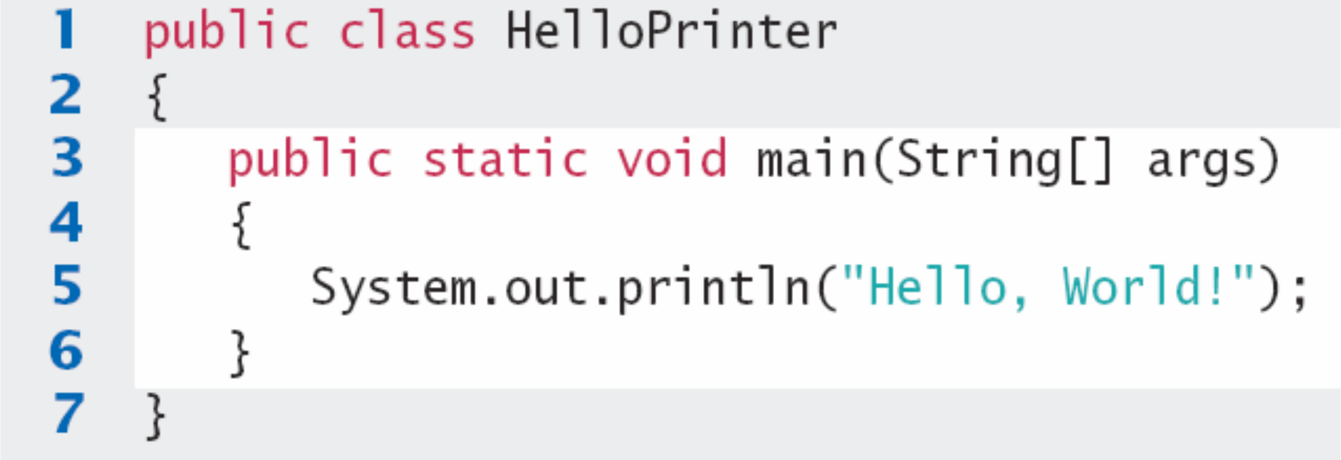
Organize your work

- Your ‘source code’ is stored in `.java` files
- Create one folder per program
  - Can be many `.java` files
- Be sure you know where your IDE stores your files!
- Backup your work!



```
ch01
  ▼ hello
      HelloPrinter.java
ch02
  ▼ initials
      Initials.java
  ▼ stamp
      StampMachine.java
  ▼ vending
```

1.5 Analyzing your First Program



```
1  public class HelloPrinter
2  {
3     public static void main(String[] args)
4     {
5        System.out.println("Hello, World!");
6     }
7  }
```

1: Declares a ‘`class`’ `HelloPrinter`

-- Every Java program has one or more classes.

3: Declares a method called ‘`main`’

-- Every Java application has exactly one ‘ `main`’ method

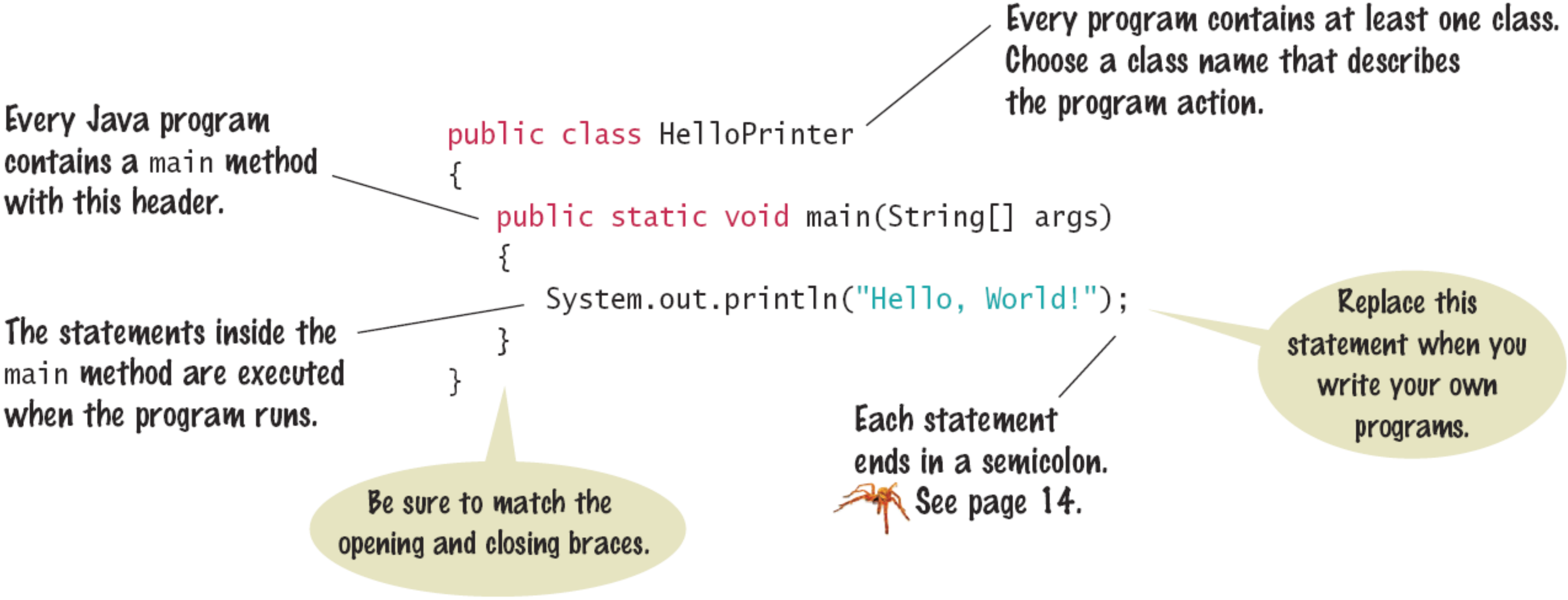
-- Entry point where the program starts

5: Method `System.out.println` outputs ‘Hello, World!’

-- A statement must end with a semicolon (`;`)

Syntax 1.1: The Java Program

- Every application has the same basic layout
  - Add your ‘code’ inside the `main` method



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

Every program contains at least one class. Choose a class name that describes the program action.

Every Java program contains a main method with this header.

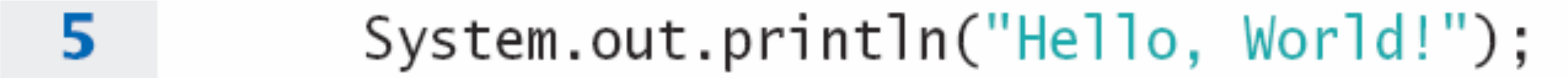
The statements inside the main method are executed when the program runs.

Replace this statement when you write your own programs.

Each statement ends in a semicolon. See page 14.

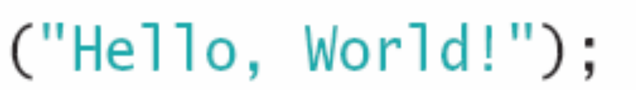
Be sure to match the opening and closing braces.

Calling Java Library methods



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

- Line 5 shows how to ‘call’ a ‘method’ from the Java API: `System.out.println`
  - Code that somebody else wrote for you!
  - Notice the dots (periods)
  - Parenthesis surround the arguments that you ‘pass’ to a method



```
("Hello, World!");
```

  - We are passing a String “Hello World”
    - Note the double quotes which denote a String inside
  - You can also print numerical values
    - `System.out.println(3 + 4);`

Getting to know println

- The println method prints a string or a number and then starts a new line.

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

```
Hello
World!
```

- println has a ‘cousin’ method named print that does not print a new line.

```
System.out.print("00");
System.out.println(3+4);
```

```
007
```

A method is called by specifying the method and its agruments

Common Error 1.1

- Omitting Semicolons
  - In Java, every statement must end in a semicolon. Forgetting to type a semicolon is a common error. It confuses the compiler, because the compiler uses the semicolon to find where one statement ends and the next one starts. For example, the compiler sees this:

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

  - As this:

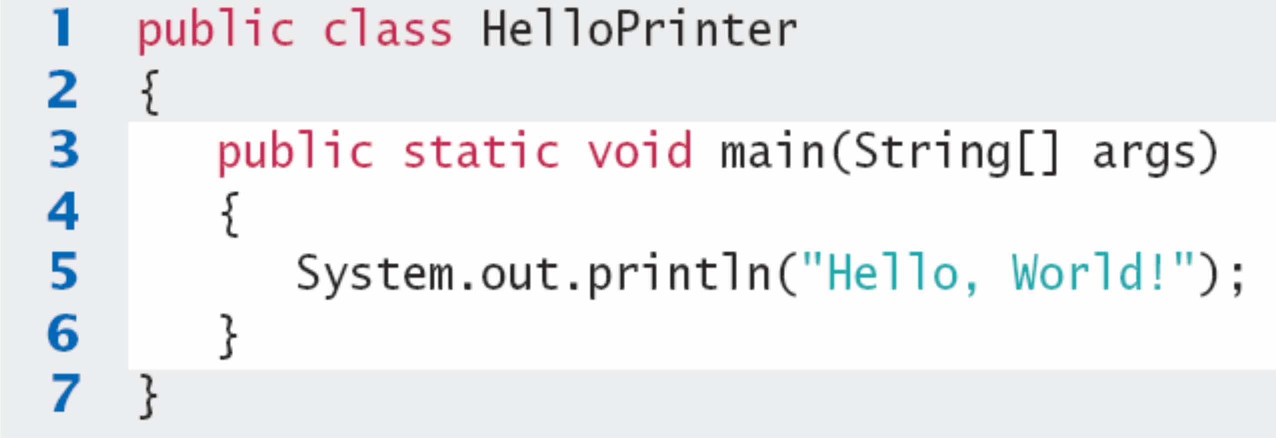
```
System.out.println("Hello") System.out.println("World!");
```

  - It doesn’t understand this statement, because it does not expect the word `System` following the closing parenthesis after `Hello`.

1.6 Errors

- The Two Categories of Errors:
  1) Compile-time Errors
     - Syntax Errors
       - Spelling, Capitalization, punctuation
       - Ordering of statements, matching of braces/parenthesis…
     - No `.class` file is generated by the compiler
     - Correct first error listed, then compile again
  2) Run-time Errors
     - Logic Errors
     - Program runs, but produces unintended results
     - Program may 'crash'

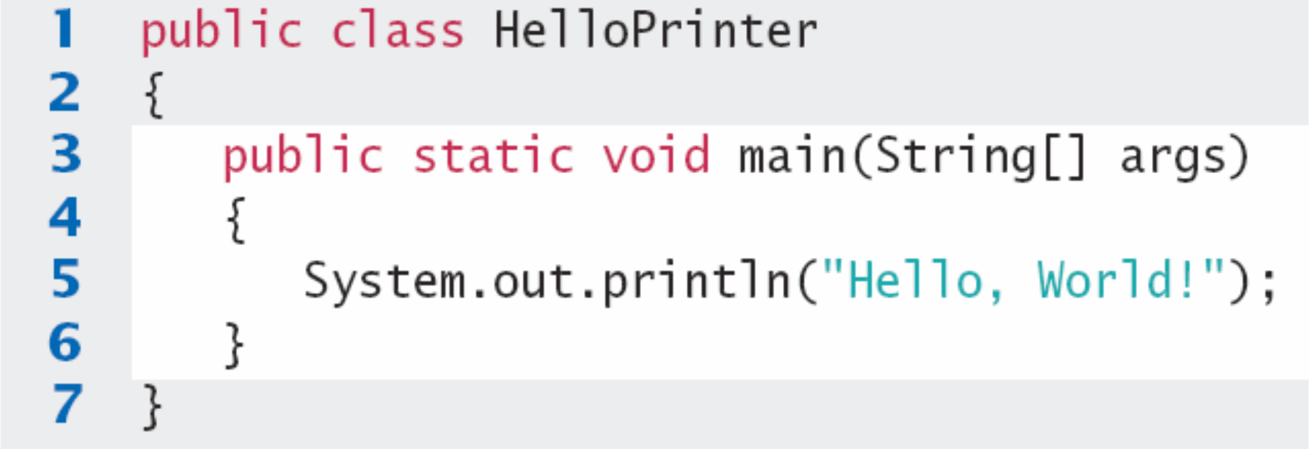
Syntax Errors



```
1  public class HelloPrinter
2  {
3     public static void main(String[] args)
4     {
5        System.out.println("Hello, World!");
6     }
7  }
```

- What happens if you
  - Misspell a word: `System.ou.println`
  - Don’t Capitalize a word `system.out.println`
  - Leave out a word `void`
  - Forget a Semicolon after `("Hello, World!")`
  - Don’t match a curly brace? Remove line 6
- Try it to see what error messages are generated

Logic Errors



```
1  public class HelloPrinter
2  {
3     public static void main(String[] args)
4     {
5        System.out.println("Hello, World!");
6     }
7  }
```

- What happens if you
  - Divide by Zero `System.out.println(1/0);`
  - Mis-spell output `("Hello, Word!")`
  - Forget to output Remove line 5
- Programs will compile and run
  - The output may not be as expected

Summary: Java

- Java was originally designed for programming consumer devices, but it was first successfully used to write Internet applets.
- Java was designed to be safe and portable, benefiting both Internet users and students.
- Java programs are distributed as instructions for a virtual machine, making them platform-independent.
- Java has a very large set of libraries. Focus on learning those parts of libraries that you need for your programming projects.

Summary: Java

- Set aside some time to become familiar with the programming environment that you will use for your class work.
- An editor is a program for entering and modifying text, such as a Java program.
- Java is case sensitive. You must be careful about distinguishing between upper and lowercase letters.
- The Java compiler translates source code into class files that contain instructions for the Java virtual machine.

Summary: Java

- Classes are the fundamental building blocks of Java programs.
- Every Java application contains a class with a main method. When the application starts, the instructions in the main method are executed.
- Each class contains declarations of methods. Each method contains a sequence of instructions.
- A method is called by specifying the method and its parameters.
- A string is a sequence of characters enclosed in quotation marks.

Summary: Errors and Pseudocode

- A compile-time error is a violation of the programming language rules that is detected by the compiler.
- A run-time error causes a program to take an action that the programmer did not intend.
- Pseudocode is an informal description of a sequence of steps for solving a problem.
- An algorithm for solving a problem is a sequence of steps that is unambiguous, executable, and terminating.