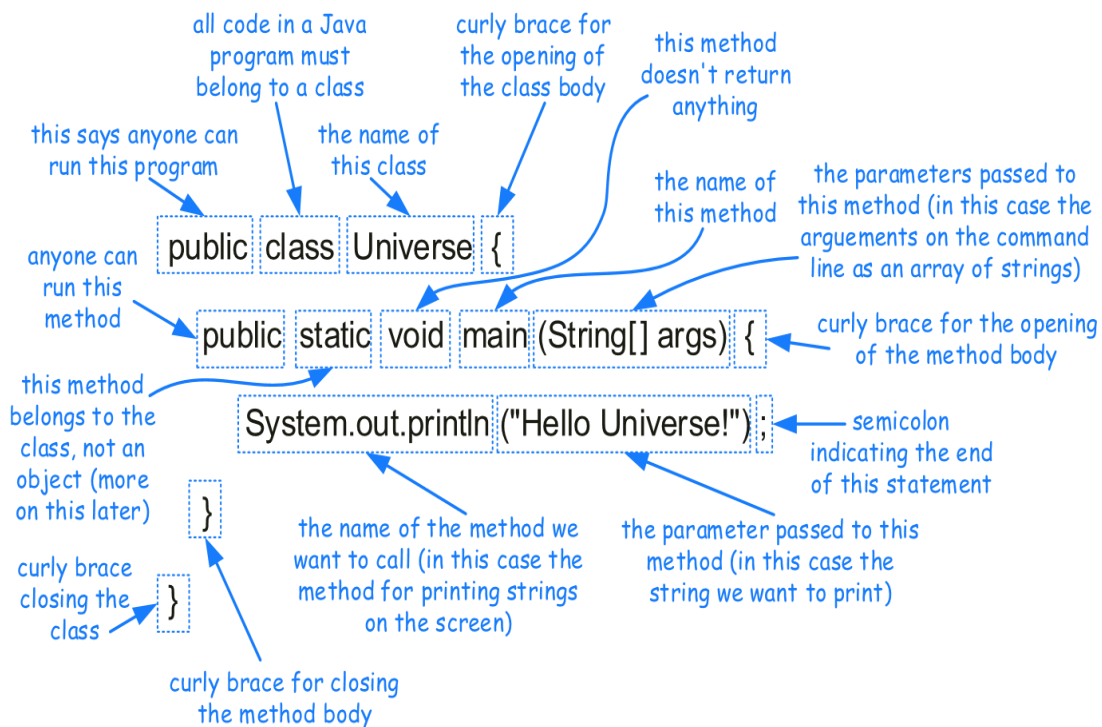


## 1.1 Getting Started



**Figure 1.1:** A “Hello Universe!” program.

In Java, executable statements are placed in functions, known as **methods** that belong to **class** definitions. Any set of statements between the braces “{” and “}” define a program **block**.

The name of a class, method, or variable in Java is called an **identifier**, which can be any string of characters as long as it begins with a letter and consists of letters, numbers, and underscore characters (where “letter” and “number” can be from any written language defined in the Unicode character set).

## 1.1. Getting Started

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

**Table 1.1:** A listing of the reserved words in Java. These names cannot be used as class, method, or variable names.

### Comments

In addition to executable statements and declarations, Java allows a programmer to embed comments, which are annotations provided for human readers that are not processed by the Java compiler. Java allows two kinds of comments: inline comments and block comments. Java uses a “//” to begin an inline comment, ignoring everything subsequently on that line. For example:

```
// This is an inline comment.
```

While inline comments are limited to one line, Java allows multiline comments in the form of block comments. Java uses a “/\*” to begin a block comment and a “\*/” to close it. For example:

```
/*  
    * This is a block comment.  
*/
```

---

### 1.1.1 Base Types

For the most commonly used data types, Java provides the following *base types* (also called *primitive types*):

**boolean** a boolean value: true or false  
**char** 16-bit Unicode character  
**byte** 8-bit signed two's complement integer  
**short** 16-bit signed two's complement integer  
**int** 32-bit signed two's complement integer  
**long** 64-bit signed two's complement integer  
**float** 32-bit floating-point number (IEEE 754-1985)  
**double** 64-bit floating-point number (IEEE 754-1985)

```
1  boolean flag = true;
2  boolean verbose, debug;           // two variables declared, but not yet initialized
3  char grade = 'A';

4  byte b = 12;

5  short s = 24;

6  int i, j, k = 257;                // three variables declared; only k initialized
7  long l = 890L;                   // note the use of "L" here
8  float pi = 3.1416F;              // note the use of "F" here
9  double e = 2.71828, a = 6.022e23; // both variables are initialized
```

**Code Fragment 1.1:** Declarations and initializations of several base-type variables.