Unit 1
The Basics of a Java Program

TABLE OF CONTENTS

1. My First Java Program

- Setting the Paths
- Using a Text Editor
- Compiling and Executing a Java Source Code
- Errors

2. The Java Program Structure

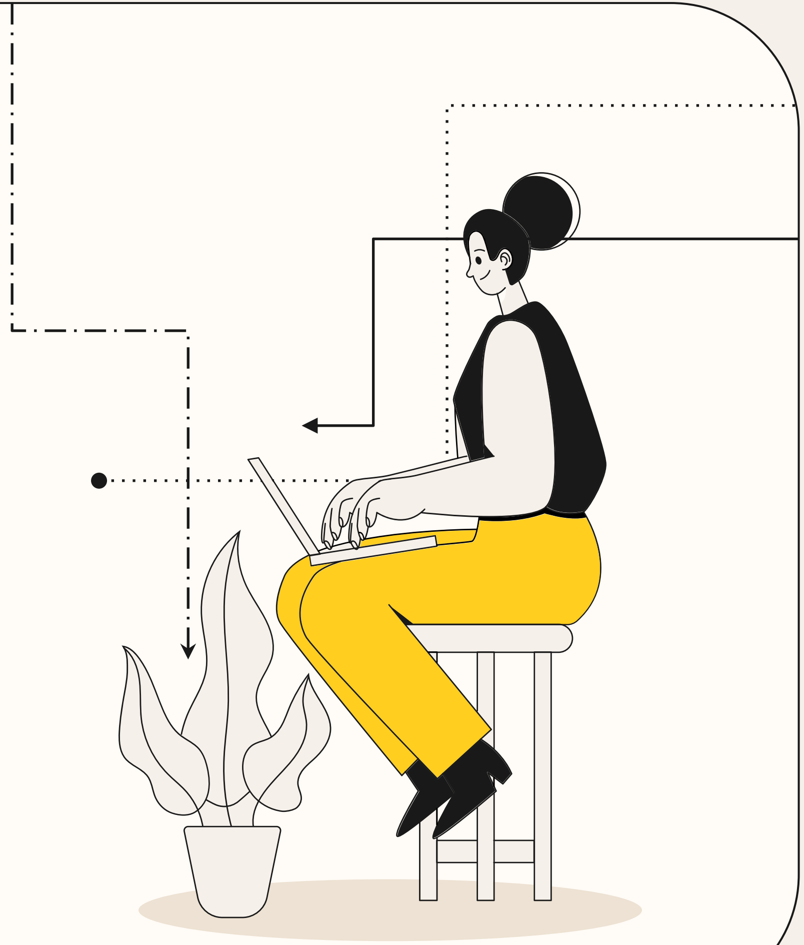
- Java Statements and Blocks
- Java Identifiers
- Java Literal and Constants
- Primitive Data Types
- String Class
- Program Variables

Did you know?

Java is used by many large companies, including Amazon, Google, and Netflix, to develop their software systems.

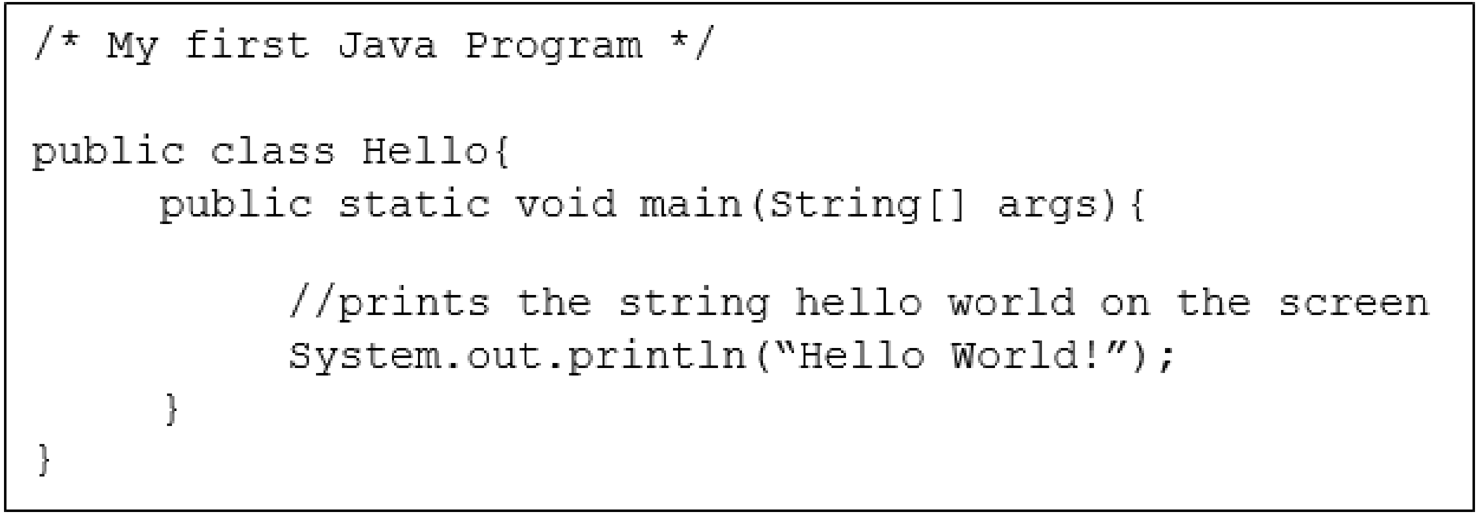
1.

My First Java Program



Program Source Code

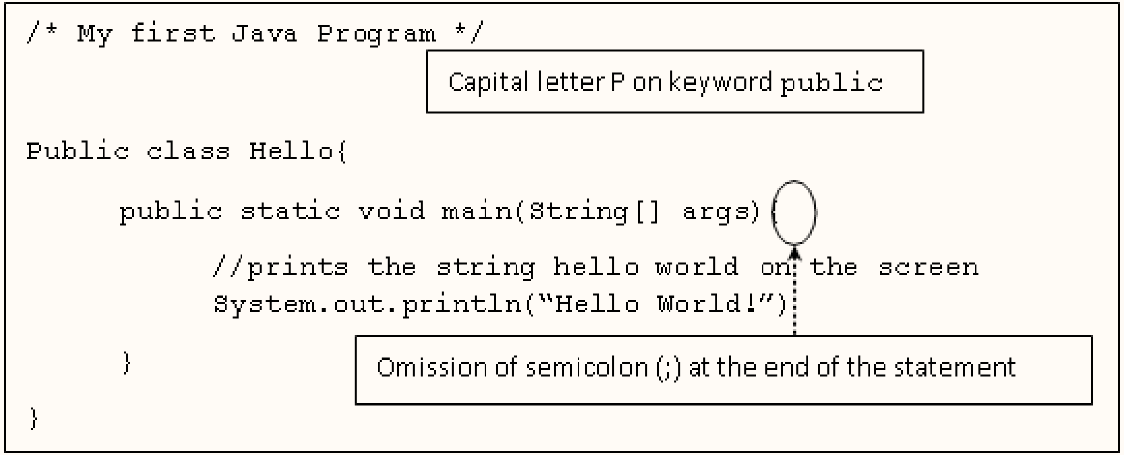
- It is a programming statement that is created by a programmer with a text editor or a visual programming tool and then saved in a file.



```
/* My first Java Program */

public class Hello{
     public static void main(String[] args){

          //prints the string hello world on the screen
          System.out.println(“Hello World!”);
     }
}
```

Types of Errors

Syntax Errors: are errors in form. These are encountered if a programmer inadvertently committed an error in typing the source code.



```
/* My first Java Program */

Public class Hello{

      public static void main(String[] args){

            //prints the string hello world on the screen
            System.out.println(“Hello World!”)

      }

}
```

Capital letter P on keyword public

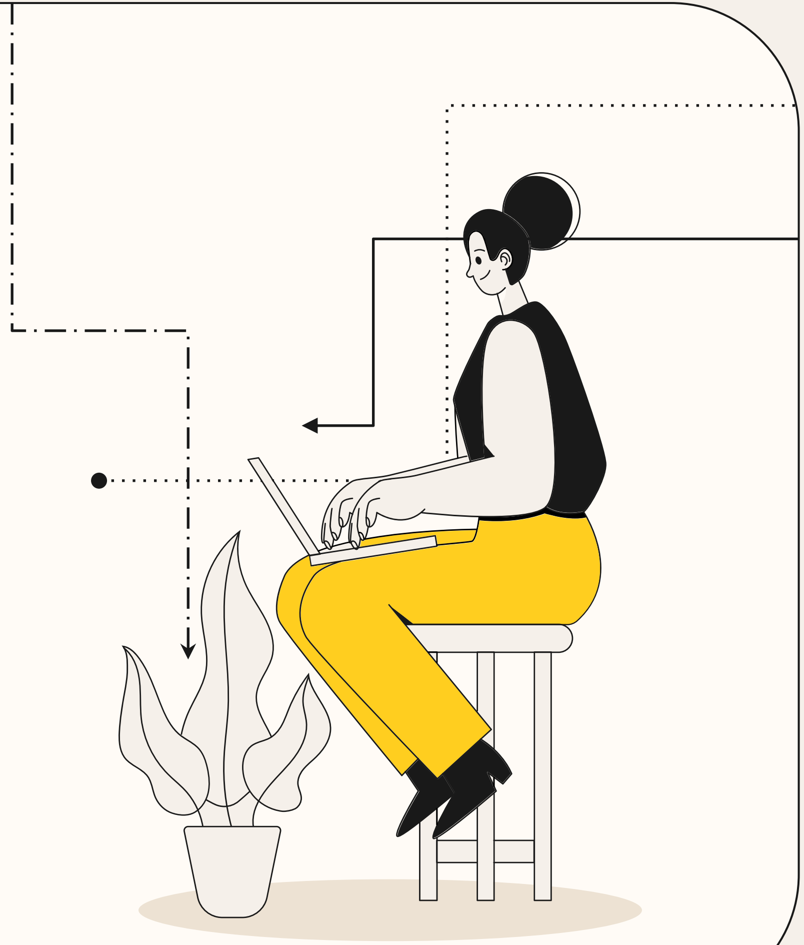
Omission of semicolon (;) at the end of the statement

Types of Errors

Run-time error: Run–time errors are errors in meaning (semantic). These are sometimes referred to as logical errors.

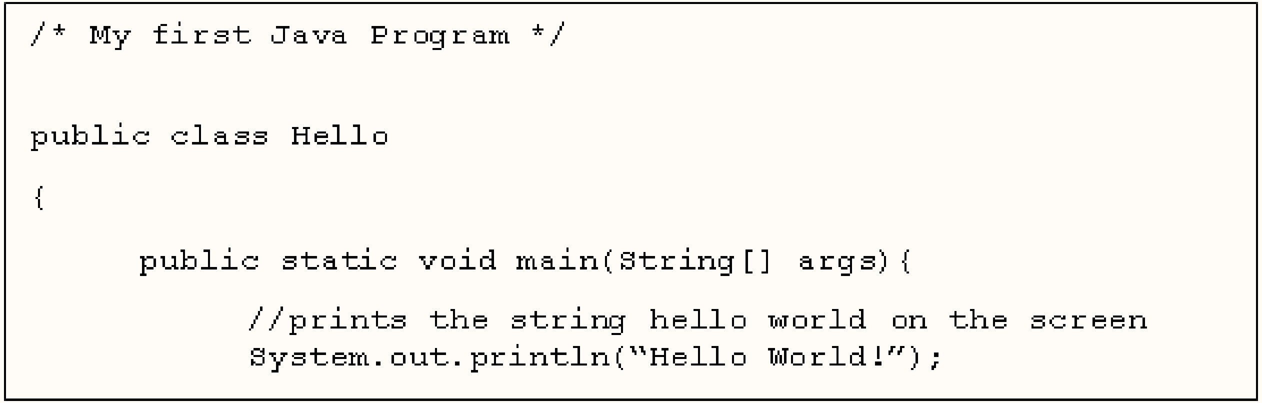
2.

The Java Program Structure



Comment/Remark

- A comment is used to document a part of the code. It is not part of the program itself and is only used for documentation to make it more readable.

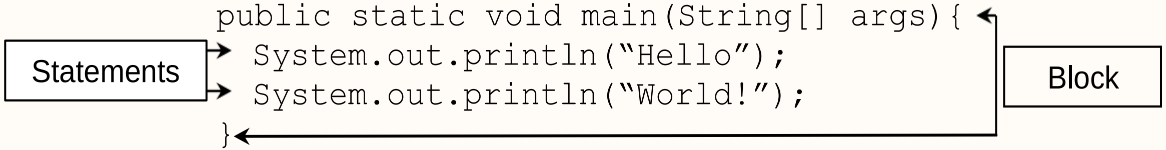


```
/* My first Java Program */


public class Hello

{

      public static void main(String[] args){

           //prints the string hello world on the screen
           System.out.println(“Hello World!”);
```

Java Statement and Blocks

The statement or an instruction is one or more lines of code terminated by a semicolon.

A block is one or more statements bounded by open and closed curly braces that group the statements as one unit.



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

1.1. Java Identifiers

Java Identifiers

- Identifiers are tokens. These are usually user-defined names that represent labels of variables, methods, and classes to name a few.

Rules for Identifiers in Java

1. The identifier should contain only

Letters [(a to z) and (A to Z)]
Digits (0 to 9)
Special characters (only $ or _)

Rules for Identifiers in Java

2. Identifiers should not start with a digit.

But a digit can be used from the second character onwards.

```
1number // invalid
n1 // valid
num1 // valid
temp122 // valid
```

Rules for Identifiers in Java

3. **Java identifiers should not contain any special character except ‘$’ and ‘_’.**

For example:- num$, number_one, num$_ are valid Java identifiers but num@, #num$_ are illegal Java identifiers.

Rules for Identifiers in Java

4. Java identifiers should not contain any space.

First Number // invalid
First_Number // valid
First$Number // valid

Rules for Identifiers in Java

5. Java identifiers are case sensitive.

The lowercase is different from the upper case.

```
int number = 10;
int NuMber = 20;
System.out.println(number); // 10
System.out.println(NuMber); // 20
```

Java Coding Convention

1. **For names of classes, capitalize the first letter of the class name.**

2. The word’s first letter should start with a small letter for method and variable names.

3. For multi-word identifiers, capitalize the first letter of each word except the first word.

Valid Identifiers

a

A

age

num1

xyz

final_grade

employee33

id_Number

firstName

sum

aReA5b3h1

Invalid Identifiers

6 Must start with a letter

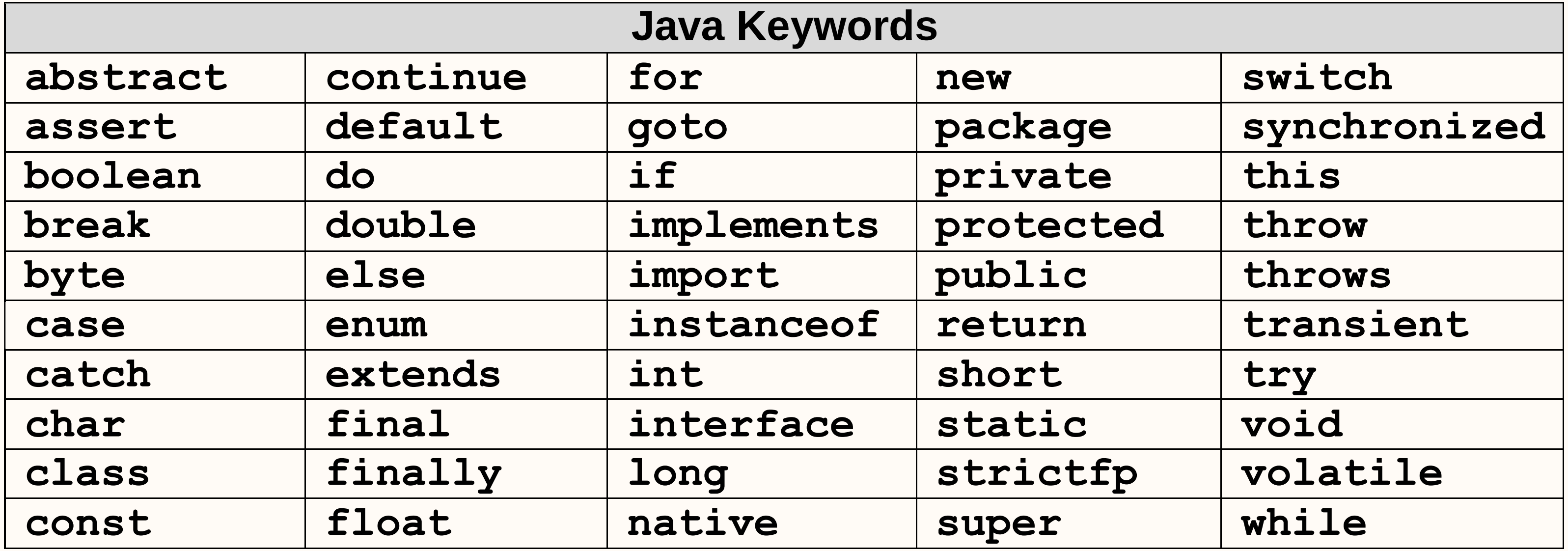
7 1st_Name Must start with a letter

8 u&me & symbol is not a letter or number

9 percent% % symbol is not a letter of a number

10 last name Space is not a letter or number

Java Keywords



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

5 Kinds of Java Literals

Literals: Literals in Java are a synthetic representation of boolean, character, numeric, or string data

1. **Integral Literals:** a numeric value(associated with numbers) without any fractional or exponential part

Decimal (base 10)	//12
Hexadecimal (base 16)	//0xC
Octal (base)	//014

5 Kinds of Java Literals

2. **Floating Point Literals:** Represent decimals with fractional parts that can be expressed instandard or scientific notations.

Example:

```
3.1416
54.567
5.8234e2
10.2000e4
```

5 Kinds of Java Literals

3. Boolean Literals

Boolean literals have only two (2) values. It is either *“True”* or *“False”.*

5 Kinds of Java Literals

4. Character Literals

Characters represent a single Unicode character.

Example:

‘a’	Letter a
‘Z’	Letter Z
‘\n’	New line character
‘\b’	Carriage return character

5 Kinds of Java Literals

5. String Literals

Represents multiple characters and are enclosed by double quotes.

Example:

```
“Hello World”
“Java Programming”
```

Primitive Data Type

- a primitive data type is a basic data type that is provided by the programming language and is not derived from any other type.

There are eight primitive data types in Java

- **byte:** A byte is an 8-bit signed integer.
- **short:** A short is a 16-bit signed integer.
- **int:** An int is a 32-bit signed integer.
- **long:** A long is a 64-bit signed integer.
- **float:** A float is a 32-bit floating-point number.
- **double:** A double is a 64-bit floating-point number.
- **boolean:** A boolean can have two possible values: true or false.
- **char:** A char is a 16-bit Unicode character.

Program Variables

- Variables are entities where data can be stored into it.
- Values stored in the variable can be changed anytime. It is an abstraction of the computer memory cell or collection of cells

Let’s try!

In your Java workbook, kindly practice and do the
2.6.1
2.6.2
2.6.3

In pages 16 to 18