Syntax:

1. Comments in Java

There are three types of comments in Java.

i. Single line Comment

```
// System.out.println("This is an comment.");
```

ii. Multi-line Comment

```
/*

    System.out.println("This is the first line comment.");

    System.out.println("This is the second line comment.");

*/
```

iii. Documentation Comment. Also called a **doc comment**.

```
/** documentation */
```

2. Source File Name

The name of a source file should exactly match the public class name with the extension of .**java**. The name of the file can be a different name if it does not have any public class. Assume you have a public class **GFG**.

```
GFG.java // valid syntax

gfg.java // invalid syntax
```

3. Case Sensitivity

Java is a case-sensitive language, which means that the identifiers ***AB, Ab, aB***, and ***ab*** are different in Java.

```
System.out.println("GeeksforGeeks"); // valid syntax

system.out.println("GeeksforGeeks"); // invalid syntax because of the
first letter of System keyword is always uppercase.
```

4. Class Names

i. The first letter of the class should be in Uppercase (lowercase is allowed but discouraged).

ii. If several words are used to form the name of the class, each inner word's first letter should be in Uppercase. Underscores are allowed, but not recommended. Also allowed are numbers and currency symbols, although the latter are also discouraged because they are used for a special purpose (for inner and anonymous classes).

```
class MyJavaProgram    // valid syntax

class 1Program         // invalid syntax

class My1Program       // valid syntax

class $Program         // valid syntax, but discouraged

class My$Program       // valid syntax, but discouraged (inner class
Program inside the class My)
```

```
class myJavaProgram    // valid syntax, but discouraged
```

5. public static void main(String [] args)

The method main() is the main entry point into a Java program; this is where the processing starts. Also allowed is the signature **public static void main(String… args)**.

6. Method Names

i. All the method names should start with a lowercase letter (uppercase is also allowed but lowercase is recommended).

ii. If several words are used to form the name of the method, then each first letter of the inner word should be in Uppercase. Underscores are allowed, but not recommended. Also allowed are digits and currency symbols.

```
public void employeeRecords() // valid syntax

public void EmployeeRecords() // valid syntax, but discouraged
```

7. Identifiers in java

Identifiers are the names of local variables, instance and class variables, and labels, but also the names for classes, packages, modules and methods. All Unicode characters are valid, not just the ASCII subset.

i. All identifiers can begin with a letter, a currency symbol or an underscore (**_**). According to the convention, a letter should be lower case for variables.

ii. The first character of identifiers can be followed by any combination of letters, digits, currency symbols and the underscore. The underscore is not recommended for the names of variables. Constants (static final attributes and enums) should be in all Uppercase letters.

iii. Most importantly identifiers are case-sensitive.

iv. A keyword cannot be used as an identifier since it is a reserved word and has some special meaning.

```
Legal identifiers: MinNumber, total, ak74, hello_world, $amount,
_under_value

Illegal identifiers: 74ak, -amount
```

8. White spaces in Java

A line containing only white spaces, possibly with the comment, is known as a blank line, and the Java compiler totally ignores it.

9. Access Modifiers: These modifiers control the scope of class and methods.

- **Access Modifiers:** default, public, protected, private.
- **Non-access Modifiers:** final, abstract, static, transient, synchronized, volatile, native.

10. Understanding Access Modifiers:

Access Modifier	Within Class	Within Package	Outside Package by subclass only	Outside Package
Private	Yes	No	No	No
Default	Yes	Yes	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

11. Java Keywords

Keywords or Reserved words are the words in a language that are used for some internal process or represent some predefined actions. These words are therefore not allowed to use as variable names or objects.

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
extends	final	finally	float
for	goto	if	implements
import	instanceof	int	interface
long	native	new	package

private	protected	public	return
short	static	strictfp	super
switch	synchronized	this	throw
throws	transient	try	void
volatile	while		