Java Basic Syntax

Difficulty Level : Easy Last Updated : 05 Mar, 2021

A Java program is a collection of objects, and these objects communicate through method calls to each other to work together. Here is a brief discussion on the Classes and Objects, Method, Instance variables, syntax, and semantics of Java.

Basic terminologies in Java

1. Object: The object is an instance of a class, have Behavior and state.

- Example: A car is an object whose **states** are: brand, colour, number-plate.
- Behavior**:** Running on the road.

2. Class: The class is a blueprint(plan) of class objects and status.

- **Example**: Blueprint of the house is class.

3. Method: The behavior of an object is the method.

- **Example**: The fuel indicator indicates the amount of fuel left in the car.

4. Instance variables: Every object has its own unique set of instance variables. The state of an object is generally created by the values that are assigned to these instance variables.

Example: Steps to compile and run a java program in a console

```
javac GFG.java
java GFG
```

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

Output

```
GFG!
```

Note: When the class is public, the name of the file should be the class name.

The Basic Syntax:

1. Comments in Java

There are three types of comments in Java.

i. Single line Comment

```
// System.out.println("GFG!");
```

ii. Multi-line Comment

```
/*
    System.out.println("GFG!");
    System.out.println("Alice!");
*/
```

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

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

2. Program File Name

The name of a program file should exactly match the class name with an extension of **.java**. The name of the file can be other names if the program 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("Alice"); // valid syntax
system.out.println("Alice"); // invalid syntax
```

4. Class Names

i. The first letter of the class should be in Uppercase

ii. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case, Underscore are allowed.

```
class MyJavaProgram    // valid syntax
class myJavaProgram    // invalid syntax
```

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

Java program processing starts with the method main().

6. Method Names

i. All the method names should start with a Lower Case letter.

ii. If several words are used to form the name of the method, then each first letter of the inner word should be in Upper Case, Underscore are allowed. (This is allowed in java and please correct the below example as well both are correct syntax as per Java but as standard people follow lowercase first char in function name)

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

7. Identifiers in java

i. All identifiers can begin with a letter (**A to Z** or **a to z**) or an underscore _.

ii. The first character of identifiers can have any combination of characters.

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, strictfp

10. 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		