

Source Code :-

Class A {

cmd [javac filename.java]

Run java <classname> }

name.pdf
4 doc
B. 'xlsx'
Test.java Source code
any name

compiler - javac

[javac Test.java] → Byte code → class

<class names

A.class

execute

JVM

java A

<class name>

1. Class Name and FileName is different - Code will be compiled and run successfully.
2. ClassName and Filename is same - Code will be compiled and run successfully.
3. Class Name and FileName is different and using public keyword infront of className: Error
4. if we are using public keyword infront of className, then classname and FileName should be same.

<u>Case 1.</u> ✓✓ Test.java Class A ✓ }	<u>Case 2.</u> ✓✓ Test.java Class Test ✓ }	<u>Case 3.</u> error X Test.java public Class A ✓ }	<u>Case 4.</u> ✓✓ A.java public Class A ✓ }
--	---	--	--

Name Convention

Class Name - always starts with uppercase - Alphabet, \$

FileName - " " " " - Alphabet

Method Name - " " " lowercase - main()

Keyword - Built-in (Already) (lowercase)

Variable - lowercase, args

Can't start with number

\$, -, letter, char

Keywords, are reserved word used for special purpose - if, else, for, do while, while

```
public class Test{  
    public static void main(String[] args){  
        System.out.print("Hello Java Code.....");  
    }  
}
```

Annotations and Identifiers in the code:

- K** (Keyword) points to `public`.
- I(c)** (Identifier) points to `Test`.
- I(m)** (Identifier) points to `main`.
- I(c)** (Identifier) points to `String`.
- I(v)** (Identifier) points to `args`.
- I(c)** (Identifier) points to `System`.
- I(prop)** (Identifier) points to `out`.
- I(m)** (Identifier) points to `print`.
- (L)** (Literal) points to the string `"Hello Java Code....."`.

— Keyword (predefined)

— Identifier

- ClassName
- varname
- method name

— Literal - any value

predefined literals

[null, true, false]

List of Java Keywords

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

Variable → Holder, Container which hold/
store the value

a = 10 ;

b = true ;

c = 'a' ;

d = "Test" ;