

# IDENTIFIERS IN JAVA

— The Names That Define Your Code

**Identifiers are names used to identify variables, classes, methods, packages, and other elements in your Java program.**

**Example:**

```
class Test  
{  
    public static void main(String[] args) {  
        int x=10;  
    }  
}  
5
```

The diagram illustrates the identifiers in the provided Java code. The identifier 'Test' is underlined with a horizontal line and labeled '1'. The identifier 'main' is underlined with a vertical line and labeled '2'. The identifier 'args' is underlined with a vertical line and labeled '3'. The identifier 'x=10' is underlined with a vertical line and labeled '4'. The number '5' is underlined with a vertical line and labeled '5'.

**Rule 1:** The only allowed characters in java identifiers are:

- 1) a to z
- 2) A to Z
- 3) 0 to 9
- 4) \_ (underscore)
- 5) \$

**Rule 2:** identifiers are not allowed to starts with digit.

Example:

- 1) ABC123-----valid
- 2) 123ABC-----invalid

**Rule 3:** java identifiers are case sensitive ofcourse java language itself treated as case sensitive language.

Example:

```
class Test{  
int number=10;  
int Number=20;  
int NUMBER=20; //we can differentiate with case.  
int NuMbEr=30;  
}
```

**Rule 4:** There is no length limit for java identifiers but it is not recommended to take more than 15 lengths.

**Rule 5:** We can't use reserved words as identifiers.

**Example:**

```
int if=10;// -----invalid
```

**Rule 6:** All predefined java class names and interface names we use as identifiers.

**Example 1:**

```
class Test
{
    public static void main(String[] args){
        int String=10;
        System.out.println(String);
    }
}
```

**Output:**

```
10
```

**Example 2:**

```
class Test
{
    public static void main(String[] args){
        int Runnable=10;
        System.out.println(Runnable);
    }
}
```

**Output:**

```
10
```

Even though it is legal to use class names and interface names as identifiers but it is not a good programming practice.