

Identifiers in Java

Identifiers are the names used in Java programs to identify elements such as classes, methods, variables, etc.

Example

```
1 // BasicSyntax.java
2 class BasicSyntax {
3     public static void main(String[] args) {
4         System.out.println("Good Night !! Friends");
5     }
6 }
```

Identifiers in the above example:



- `BasicSyntax` — User-defined class name
- `System`, `String` — Predefined classes from the Java API
- `main()`, `println()` — Predefined methods
- `args` — User-defined variable (parameter)
- `out` — Predefined variable (member of `System` class)

Rules for Writing Identifiers



1. Allowed Characters:

- Lowercase alphabets: `a-z`
- Uppercase alphabets: `A-Z`
- Digits: `0-9`
- Special characters: `$` and `_`




2. Starting Character Restrictions:

- Identifiers **must not start** with a digit.
 -  `Sambit22`
 -  `22Sambit`




3. Reserved Keywords:

- Reserved Java keywords **cannot** be used as identifiers.
 -  `class Sambit { ... }`
 -  `class static { ... }`, `class void { ... }`, `class if { ... }`

4. Case Sensitivity:

- Java identifiers are **case-sensitive**.
 -  `Sambit`, `sambit` — different identifiers
 -  `Switch` (valid identifier),  `switch` (keyword)

5. Special Notes:

- A standalone underscore (`_`) is **not allowed** as of Java 9.
 -  `_name`
 -  `_`
- **Spaces are not permitted** in identifiers.
 -  `class Sambit Sagar { ... }`

ASCII Values Reference

ASCII (American Standard Code for Information Interchange) assigns numeric codes to characters:

- Uppercase: `A-Z` → `65-90`
- Lowercase: `a-z` → `97-122`
- Digits: `0-9` → `48-57`

Case Sensitivity Example:

Character	ASCII Decimal	ASCII Binary
I	73	01001001
i	105	01101001
f	102	01100110

`If` and `if` are different due to their distinct ASCII values.

Camel Case Convention

Java follows the **Camel Case** convention for naming identifiers:

- No spaces or separators.
- The first word starts with a **lowercase** letter.
- Each subsequent word starts with an **uppercase** letter.

Examples:

- `functionName`
 - `longerFunctionName`
 - `evenLongerFunctionName`
-