**Java Identifiers**

Identifiers are the names of variables, methods, classes, packages and interfaces. Unlike literals they are not the things themselves, just ways of referring to them. In the HelloWorld program, **HelloWorld**,  **args**, **main** and**println** are identifiers.

Identifiers must be composed of letters, numbers, the underscore \_ and the dollar sign $. Identifiers may only begin with a letter, the underscore or a dollar sign.

Java uses CamelCase as a practice for writing names of Identifiers (methods, variables, classes, packages and constants)

**Camel case consists** of compound words or phrases such that each word or abbreviation begins with a capital letter or first word with a lowercase letter, rest all with capital.

**Java Modifiers**

Modifiers are keywords that you add to those Identifiers to change their meanings. Java language has a wide variety of modifiers, including the following −

* Java Access Modifiers
* Non-Access Modifiers

To use a modifier, you include its keyword in the definition of a class, method, or variable. The modifier precedes the rest of the statement, as in the following example, the words in bold blue are modifiers.

In Java, identifiers are names given to various program elements such as variables, methods, classes, packages, etc. Here are the rules for defining Java identifiers:

1. Must Begin with a Letter, Currency Character ($), or Underscore (\_):

- A Java identifier must start with a letter (uppercase or lowercase), currency character (`$`), or underscore (`\_`).

2. Followed by Letters, Digits, Currency Characters, or Underscores:

- After the initial character, a Java identifier can consist of letters, digits, currency characters, or underscores.

3. No Spaces or Special Characters (Except Currency Characters and Underscores):

- Spaces and special characters (except for `$` and `\_`) are not allowed in identifiers.

4. No Reserved Words:

- Identifiers cannot be the same as Java reserved words (keywords).

5. Case-Sensitive:

- Java is case-sensitive, so `myVariable` and `myvariable` are considered different identifiers.

6. No Length Limitation:

- There is no specific limitation on the length of identifiers in Java, but it's good practice to keep them reasonably short and meaningful.

### Examples of Valid Identifiers:

```java

int myVariable;

String userName;

double $amount;

int \_value;

```

Examples of Invalid Identifiers:

```java

// Invalid: starts with a digit

int 123variable;

// Invalid: contains a special character

double my$Variable;

// Invalid: uses a reserved word

boolean class;

```

In Java, it's crucial to choose meaningful and descriptive identifiers to enhance the readability and maintainability of your code. Following these rules ensures that your identifiers are valid according to the Java language specifications.