# Java Naming Conventions and Valid Identifiers

## 1. Introduction

Naming conventions in Java are essential for writing clean, readable, and maintainable code. Java provides flexibility in naming identifiers, but following standard conventions helps in maintaining consistency across projects.

---

## 2. Class Naming Conventions

**Allowed:**

- Must begin with a **letter (A–Z or a–z)**, **underscore (_)**, or **dollar sign ($)**

- Can include **letters, digits (0–9)**, **underscores (_)**, and **dollar signs ($)**

- **Cannot begin with a digit**

- **Cannot be a reserved Java keyword**

**Conventions:**

- Use **PascalCase** (Each word starts with a capital letter)

- Name should be a **noun**

**Examples:**

```
class Student {}
class BankAccount {}
class MyClass123 {}
class $MainClass {}
```

**Invalid Examples:**

```
class 123Class {}    // Starts with a digit
class class {}       // 'class' is a reserved keyword
class My-Class {}    // Hyphen is not allowed
```

# 3. Package Naming Conventions

**Allowed:**

- Must start with a **letter**

- Can contain **letters, digits, and periods (.)**

- No **underscores, dollar signs, or special characters**

- Use **only lowercase letters**

**Conventions:**

- Use lowercase, domain-style format (e.g., `com.example.app`)

**Examples:**

package com.myapp;
package org.example.utils;

**Invalid Examples:**

package MyPackage;       // Uppercase not standard
package 123utils;        // Cannot start with digit
package my-package;      // Hyphens not allowed

# 4. Variable Naming Conventions

**Allowed:**

- Must begin with a **letter, underscore (_)**, or **dollar sign ($)**

- Can include **letters, digits, underscores, and dollar signs**

- Cannot be a Java keyword

**Conventions:**

- Use **camelCase**

- Name should be **descriptive nouns**

## Examples:

int age;
String studentName;
double _balance;
boolean $isValid;

## Invalid Examples:

int 2marks;          // Starts with a digit
int class;           // 'class' is a keyword
int student-name;    // Hyphens not allowed

---

# 5. Method Naming Conventions

## Allowed:

- Follows same rules as variable naming

- Must begin with **a letter, _ or $**

- Cannot be a reserved keyword

## Conventions:

- Use **camelCase**

- Method names should be **verbs or verb phrases**

## Examples:

void calculateTotal() {}
int getMarks() {}
boolean isAvailable() {}

## Invalid Examples:

void 123print() {}      // Starts with digit

```
void void() {}        // 'void' is a keyword
void print-message() {} // Hyphen not allowed
```

---

# 6. Summary Table

| Element | Convention | Example | Invalid Example |
|---|---|---|---|
| Class | PascalCase | `StudentRecord` | `123Class` |
| Package | lowercase.words | `com.example` | `MyPackage` |
| Variable | camelCase | `studentAge` | `int` (keyword) |
| Method | camelCase (verb) | `calculateMarks()` | `123print()` |

---

Following these conventions ensures that your code adheres to industry standards, improves collaboration, and enhances maintainability.