

Identifiers in Java - Full Notes

1. What is an Identifier?

An identifier is the name used to identify classes, interfaces, methods, variables, labels, and packages.

Example:

```
int age = 25; // 'age' is an identifier
```

2. Rules for Naming Identifiers

- Can use letters (AZ, az), digits (09), underscore (_) and dollar sign (\$)
- Cannot start with a digit (e.g., 1value is invalid)
- Cannot use Java keywords/reserved words (e.g., int, class)
- Case-sensitive (e.g., MyVar myvar)
- No special characters allowed (e.g., @, #, !)

3. Naming Conventions (Best Practices)

- Class: PascalCase (e.g., StudentDetails)
- Interface: PascalCase (e.g., Runnable)
- Method: camelCase (e.g., calculateTotal())
- Variable: camelCase (e.g., studentName)
- Constant: UPPERCASE_WITH_UNDERSCORES (e.g., MAX_SPEED)
- Package: lowercase (e.g., com.company.project)

4. Invalid Identifier Examples

- int: Reserved keyword
- 2ndNumber: Starts with digit
- user-name: Contains illegal character '-'

- @value: Contains illegal character '@'
- null: Reserved literal

5. Examples

```
class MyClass {  
  
    int rollNumber = 101;  
  
    String studentName = "Alice";  
  
    void displayDetails() {  
  
        System.out.println(studentName);  
  
    }  
  
}
```

6. Reserved Keywords (Cannot be used as identifiers)

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

Note: 'const' and 'goto' are reserved but not used.

7. Summary

- Begins with a letter, '_', or '\$' Valid
- Contains digits after first character Valid
- Starts with digit Invalid
- Uses Java keyword Invalid
- Contains special character like '@' Invalid
- Case-sensitive Valid