Java Keywords and Their Uses

abstract: Defines abstract classes or methods that must be implemented by subclasses.

assert: Used for debugging to test assumptions in code.

boolean: Defines a variable type with true or false values.

break: Exits from a loop or switch statement.

byte: Defines a variable of 8-bit integer type.

case: Defines a branch in a switch statement.

catch: Handles exceptions in a try block.

char: Defines a variable that stores a single character.

class: Defines a class in Java.

const: Reserved keyword, not used.

continue: Skips current loop iteration and continues to next iteration.

default: Specifies a block of code that runs if no case matches in a switch.

do: Executes a block of code once, then repeats while condition is true.

double: Defines a variable for double-precision floating-point numbers.

else: Specifies a block to execute if an if condition is false.

enum: Defines a set of constant values (enumeration).

extends: Indicates inheritance between classes.

final: Prevents inheritance, modification, or method overriding.

finally: Executes code after a try/catch block regardless of exceptions.

float: Defines a variable for single-precision floating-point numbers.

for: Creates a for loop for iteration.

goto: Reserved keyword, not used in Java.

if: Executes a block if a specified condition is true.

implements: Indicates that a class implements an interface.

import: Imports other packages or classes.

instanceof: Tests if an object is an instance of a specific class.

int: Defines a variable of integer type (32-bit).

interface: Declares an interface in Java.

long: Defines a variable of long integer type (64-bit).

native: Specifies that a method is implemented in native code (C/C++).

new: Creates a new object instance.

null: Represents the null reference (no object).

package: Defines a namespace for classes.

private: Access modifier for class members accessible only within the class.

protected: Access modifier for members accessible within the package and subclasses.

public: Access modifier allowing access from any class.

return: Exits a method and optionally returns a value.

short: Defines a variable of short integer type (16-bit).

static: Indicates that a variable or method belongs to the class, not instances.

strictfp: Ensures consistent floating-point calculations across platforms.

super: Refers to the superclass of the current object.

switch: Selects one block of code to execute based on a value.

synchronized: Prevents thread interference by locking methods or blocks.

this: Refers to the current object instance.

throw: Used to throw an exception.

throws: Declares the exceptions a method can throw.

transient: Prevents a variable from being serialized.

try: Defines a block to test for exceptions.

void: Specifies that a method does not return any value.

volatile: Indicates a variable may change unexpectedly (used in multithreading).

while: Executes a block repeatedly while a condition is true.

true: Boolean literal representing truth.

false: Boolean literal representing falsehood.