

Java Programming Cheat Sheet

Complete Edition | Calibri + Consolas | Colorized examples

Nikhil Learn Hub | nikhillearnhub.com

Code color legend

keyword string number comment @annotation

1. BASIC STRUCTURE (Hello World)

Part	Role
package / import	Namespace / bring types
public class Name	Entry class (file = Name.java)
public static void main	Program entry point
System.out.println	Print to console

Syntax

```
public class Main {  
    public static void main(String[] args) {  
        // statements  
    }  
}
```

Short example

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello Java!");  
    }  
}
```

Key points

- Compile: javac Main.java Run: java Main

2. DATA TYPES & VARIABLES

Kind	Examples	Notes
Primitive	byte short int long	Whole numbers
Floating	float double	Decimals
Other prim	char boolean	Char / true-false
Reference	String, arrays, objects	Heap objects

Syntax

```
type name = value;
final type CONST = value;
```

Short example

```
int age = 21;
double pi = 3.14;
boolean ok = true;
String name = "Ada";
final int MAX = 100;
```

3. ACCESS MODIFIERS

Modifier	Same class	Package	Subclass	World
public	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	No
(default)	Yes	Yes	No	No
private	Yes	No	No	No

Syntax

```
public class A { private int x; }
```

Short example

```
public class Box {
    private int w;
    public int getW() { return w; }
}
```

4. OPERATORS

Type	Operators	Notes
Arithmetic	+ - * / %	Math
Relational	< > <= >= == !=	Compare
Logical	&& !	Short-circuit
Bitwise	& ^ ~ << >> >>>	Bits
Assignment	= += -= *= /=	Store
Other	?: instanceof new .	Misc

Syntax

```
a + b; a && b; x instanceof String;
```

Short example

```
int a = 5, b = 2;
int m = (a > b) ? a : b;
System.out.println(m + " " + (a % b));
```

5. CONTROL FLOW

Form	Use
if / else if / else	Branching
switch / case	Multi-way on value
?:	Ternary expression

Syntax

```
if (cond) { }
else if (cond2) { }
else { }
switch (x) { case 1: break; default: break; }
```

Short example

```
int x = 10;
if (x > 0)
    System.out.println("positive");
else
    System.out.println("not positive");
```

6. LOOPS

Loop	Notes
for	init; cond; update
while	Entry-controlled
do-while	Runs at least once
for-each	for (T x : collection)
break / continue	Exit / next

Syntax

```
for (int i = 0; i < n; i++) { }
for (int x : arr) { }
```

Short example

```
for (int i = 1; i <= 3; i++)
    System.out.print(i + " ");
int[] a = {10, 20};
for (int v : a)
    System.out.print(v);
```

7. METHODS

Feature	Notes
Signature	mods ret name(params)
static	Belongs to class
return	Send value / exit
Overloading	Same name, different params
varargs	type... name

Syntax

```
modifiers returnType name(params) {
    return value;
}
```

Short example

```
static int add(int a, int b) {
    return a + b;
}
System.out.println(add(2, 3));
```

8. INPUT / OUTPUT

API	Use
System.out.print / println	Console output
Scanner	Read from System.in
BufferedReader	Efficient text input
printf	Formatted output

Syntax

```
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
```

Short example

```
import java.util.Scanner;
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
System.out.println("n=" + n);
sc.close();
```

9. ARRAYS

Form	Example
1D	int[] a = {1, 2, 3};
Size	new int[5];
2D	int[][] m = new int[2][3];
Length	a.length

Syntax

```
type[] name = new type[size];
```

Short example

```
int[] a = {10, 20, 30};
a[1] = 99;
for (int i = 0; i < a.length; i++)
    System.out.print(a[i] + " ");
```

10. STRINGS

Method	Role
length()	Char count
charAt(i)	Char at index
substring(a,b)	Slice
equals / compareTo	Compare content
toUpperCase / trim	Transform
+ / concat	Join

Syntax

```
String s = "text";
s.length();
```

Short example

```
String s = "Java";
System.out.println(s.length() + " " + s.toUpperCase());
System.out.println(s.equals("Java"));
```

Key points

- String is immutable. Use StringBuilder for heavy edits.

11. OOP - CLASS & OBJECT

Part	Meaning
class	Blueprint
object / new	Instance on heap
constructor	Initialize object
this	Current instance
static	Class-level member

Syntax

```
class Name {
    Type field;
    Name(...) { }
    Ret method(...) { }
}
```

Short example

```
class Box {
    int w;
    Box(int w) { this.w = w; }
    int area() { return w * w; }
}
Box b = new Box(4);
System.out.println(b.area());
```

12. INHERITANCE & POLYMORPHISM

Idea	Notes
extends	Single class inheritance
implements	Interfaces (multi)
@Override	Mark overridden method
abstract	Incomplete type / method
super	Parent constructor / members

Syntax

```
class Dog extends Animal { }
interface Flyable { void fly(); }
```

Short example

```
class Animal {
    void speak() { System.out.println("?"); }
}
class Dog extends Animal {
    @Override
    void speak() { System.out.println("woof"); }
}
```

13. INTERFACES & ABSTRACT

Feature	abstract class	interface
Instance fields	Yes	Constants mainly
Constructors	Yes	No
Multiple inherit	No	Yes (implements)
Methods	Abstract + concrete	Abstract / default / static

Syntax

```
abstract class Shape { abstract double area(); }
```

Short example

```
interface Printable {
    void print();
}
class Doc implements Printable {
    public void print() { System.out.println("doc"); }
}
```

14. EXCEPTION HANDLING

Keyword	Role
try	Guard risky code
catch	Handle exception
finally	Always runs
throw	Raise exception
throws	Declare checked exceptions

Syntax

```
try { } catch (Exception e) { } finally { }
```

Short example

```
try {
    int x = Integer.parseInt("x");
} catch (NumberFormatException e) {
    System.out.println("bad number");
} finally {
    System.out.println("done");
}
```

15. COLLECTIONS FRAMEWORK

Interface	Common class	Use
List	ArrayList, LinkedList	Ordered, index
Set	HashSet, TreeSet	Unique
Map	HashMap, TreeMap	Key-value
Queue	LinkedList, PriorityQueue	FIFO / priority

Syntax

```
List<Integer> list = new ArrayList<>();
```

Short example

```
import java.util.*;
List<Integer> nums = new ArrayList<>();
nums.add(10); nums.add(20);
System.out.println(nums.get(0));
```

16. GENERICS

Idea	Notes
<T>	Type parameter
List<String>	Parameterized type
Avoids casts	Compile-time type safety
Wildcard ?	Unknown type bound

Syntax

```
class Box<T> { private T value; }
```

Short example

```
class Box<T> {
    private T value;
    void set(T v) { value = v; }
    T get() { return value; }
}
Box<String> b = new Box<>();
b.set("Hi");
```

17. MULTITHREADING (QUICK)

Approach	Notes
extends Thread	Override run()
implements Runnable	Preferred composition
start()	Begin thread (not run())
synchronized	Mutual exclusion

Syntax

```
new Thread(() -> { ... }).start();
```

Short example

```
Runnable task = () ->
    System.out.println("hello thread");
new Thread(task).start();
```

18. MATH, TYPECAST, COMMANDS

Topic	Examples
Math	Math.max, Math.sqrt, Math.abs
Casting	(int) 3.9 // 3
Autoboxing	Integer x = 5;
Commands	javac File.java java File

Syntax

```
double r = Math.sqrt(9);
int n = (int) 3.9;
```

Short example

```
System.out.println(Math.max(3, 7));
double d = 9.7;
int n = (int) d; // 9
Integer boxed = n;
```

19. REGEX (QUICK)

API	Role
String.matches	Full match
replaceAll	Replace by pattern
Pattern / Matcher	Advanced matching

Syntax

```
s.matches("\\d+");
s.replaceAll("\\s+", " ");
```

Short example

```
String s = "a12b34";
System.out.println(s.replaceAll("\\D", "")); // 1234
```

20. INTERVIEW HOTSPOTS

Topic	Revise
OOP 4 pillars	Encapsulation, inheritance, poly, abstraction
== vs equals	Reference vs content
String immutability	StringBuilder / StringBuffer
Collections choice	List vs Set vs Map complexity
Checked vs unchecked	Exception hierarchy
JVM memory	Heap, stack, GC basics
HashMap internals	hash, buckets, equals contract

Syntax

```
String a = "hi"; String b = new String("hi");  
a == b; a.equals(b);
```

Short example

```
String a = "hi";  
String b = new String("hi");  
System.out.println(a == b); // false often  
System.out.println(a.equals(b)); // true
```

Educational summary for Nikhil Learn Hub. Topic map inspired by GeeksforGeeks Java Cheat Sheet; wording is original.