

Java Core Cheat Sheet

Comprehensive Guide For Java Programming

by [Pushpender Singh \(github.com/Technowlogy-Pushpender\)](https://github.com/Technowlogy-Pushpender)

Java Programming

High Level, Object Oriented programming language developed by **James Gosling** in 1991 & is released by **Sun Microsystems** in 1996 & is currently owned by **Oracle**. Used in developing multi platform softwares.

HelloWorld Program

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

Comments In Java

Single Line Comment:

Starts with double slashes (//)

Multiple Line Comment:

Starts with /* and ends with */

Print Methods In Java

```
// Prints In New Line  
System.out.println()
```

```
// Prints IN Same Line  
System.out.print()
```

Variables In Java

Declaring (Creating) Variables

Syntax : <dataType> <variableName> = <value>;

e.g. String authorName = "Pushpender";

General Rules for Constructing Variable Names

1. Should Starts with letter, \$ (dollar) or _ (underscore)
2. Should Not Contains Whitespace
3. Should Not Contains Special Character
4. Keywords can't be used as VariableName
5. VariableNames are Case-Sensitive
 |_ that means **Name** is not equals to **name**

Creating A Constant or Final Variable (Immutable)

Syntax: final <datatype> <variableName> = <value>;
e.g. final String authorName = "Pushpender";

// You can't assign/overwrite the value of a final variable

Declaring Some Other Variables of Different Type

```
String authorName = "Pushpender";  
int age = 20;  
float percentage = 93.7f;  
char section = 'C';  
boolean isJavaFun = true;  
byte rollno = 26;  
double piValue = 3.141592653589793238;  
long populationOfIndia = 1352600000;
```

DataTypes In Java

Two Types of DataTypes

1. Primitive Data Types
2. Non-Primitive or Reference Data Types

1. Primitive Data Types

byte	Size: 1 bytes
short	Size: 2 bytes
int	Size: 4 bytes
long	Size: 8 bytes
float	Size: 4 bytes
double	Size: 8 bytes
boolean	Size: 1 bit
char	Size: 2 bytes

Non-Primitive/Reference DataTypes

String, Arrays and Classes

Operators In Java

1. **Arithmetic:** +, -, *, /, %, ++, --
2. **Assignment:** =, +=, -=, /=, *=, &=, etc
3. **Comparison:** ==, !=, >, <, >=, <=
4. **Logical:** &&, ||, !
5. **Bitwise:** ^, &, |

Type Casting In Java

There are 2 types of Casting In Java

1. **Widening Casting** (automatically) : Smaller To Larger Type.
byte -> short -> char -> int -> long -> float -> double
2. **Narrowing Casting** (manually) : Large to Small.
double -> float -> long -> int -> char -> short -> byte

1. Widening Casting (automatically)

```
int myInt = 9;  
double myD = myInt; // int --> double
```

2. Narrowing Casting (manually)

```
double myDouble = 9.78;  
int myInt = (int) myDouble; // double to int
```

Method to Convert Numeric Values to String

```
String str = String.valueOf(value);
```

Method to Convert String to Numeric Values

```
int i = Integer.parseInt(str);  
double d = Double.parseDouble(str);
```

Taking User Input In Java

```
// Using Scanner : You Have to Import it Your Program  
import java.util.Scanner;  
Scanner scan = new Scanner(System.in);  
  
String str = scan.nextLine();  
String str = scan.next(); // Takes Input Till 1st Whitespace  
  
double myDouble = scan.nextDouble();  
int num = scan.nextInt();
```

Using Console

```
String name = System.Console().readLine();
```

Strings & Its Methods/Function In Java

```
// String Concatenation : Combination of String  
String firstName = "Pushpender", lastName = "Singh";  
System.out.println("Hello " + firstName + " " + lastName);  
// Result : Hello Pushpender Singh  
  
// Useful String Methods  
String text = "Hello World";  
System.out.println(text.length); // Outputs 11  
System.out.println(text.toUpperCase); // Outputs "HELLO WORLD"  
System.out.println(text.indexOf("World")); // Outputs 7  
System.out.println(text.contains("Hello")); // Outputs true  
System.out.println(text.charAt(3)); // Outputs 'l' (Return char)  
System.out.println(text.endsWith("lol")); // Outputs false  
System.out.println(text.indexOf("Pushpender")); // Outputs -1
```

Escape Sequence in Java

Escape character	Result
\'	'
\"	"
\\	\
\\n	New Line
\\r	Carriage Return
\\t	Tab
\\b	Backspace

Math (Useful Built-In Java Class)

```
int num1 = 4, num2 = 7;  
System.out.println(Math.max(num1, num2)); // Outputs 7  
System.out.println(Math.min(num1, num2)); // Outputs 4  
System.out.println(Math.sqrt(36)); // Outputs 6.0  
System.out.println(Math.abs(-36)); // Outputs 36  
System.out.println(Math.random()); // Random No. b/w 0 & 1  
System.out.println(5+(10-5)*Math.random()); // R. No. b/w 5 & 10
```

Conditional Statements (if, else if, else)

```
int age = 34; // Example Program  
if (age < 12) { // if (condition) {expression}  
    System.out.println("You are a Kid!"); }  
  
else if (12 < age && age <= 19) { //else if (condition) {expression}  
    System.out.println("You are a Teenager"); }  
else if (19 < age && age <= 50) { //else if (condition) {expression}  
    System.out.println("You are a Adult"); }  
  
else { //else {expression}  
    System.out.println("You are a Senior Citizen!"); }  
}
```

Conditional Statements (switch statement)

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block }  
}
```

Loops In Java (Iterative Statements)

```
// for loop  
for (statement1; statement2; statement3) {expression}  
// statement1 : To Be Executed Initially for once  
// statement2 : Condition To Be Check On Each Iteration  
// statement3 : To Be Executed Every Iteration  
// for each loop  
for (<dataType> values: someArray) {}
```

```
// while loop  
while (condition) {expression}  
  
// do while loop  
do {expression} while (condition)
```

break & continue Keywords In Java

break; // Breaks the Loop and Come Out
continue; // Skip the current iteration and move forward

1. Save	className.java	2. Compile	javac className.java
3. Run	java className		

Array In Java

```
// 1-Dimensional Array  
<datatype> [] <variableName> = {value1, value2, ...};  
e.g. int [] num = {1,2,3,4,5};  
  
// Updating Arrays Index Value  
<arrayName>[indexNumber] = <newValue>;  
num[3] = 455;  
  
// n-Dimensional Array  
<type> [][] <name> = {{val1, val2,...}, {val3, val4,...}};  
String [][] cars = {{ "BMW", "Ferrari", "Tesla", "Fords" }};  
  
// Retriving particular value from 1-Dimensional Array  
System.out.println(num[0]);  
  
// Retriving particular value from 2-Dimensional Array  
System.out.println(cars[0][1]);
```

Try-catch In Java

```
// Try-catch example program  
String [] cars = {"Ferrari", "BMW", "Jaguar"};  
try {  
    System.out.println(cars[3]); }  
catch (Exception e) {  
    System.out.println(e); }  
}
```