

Become a Certified Professional➔

Core Java Cheat Sheet – Basics Of Java Programming

Last updated on Nov 29,2022 141.9K Views


Share





Swatee Chand


Sr Research Analyst at Edureka. A techno freak who likes to explore...


Java Certification Training

 Real-life Projects

 Lifetime Access

 24x7 Tech Support

 Hands-on Assessments

 **edureka!**

EXPLORE COURSE

Are you an aspiring *Java developer*? Well, if you are, then I bet you can make use of this **Java Cheat Sheet**. **Java** is known for its pre-built classes and libraries and sometimes, keeping a track of them becomes a little tricky. So, here I bring you the Core Java Cheat Sheet.

This cheat sheet will act as a crash course for Java beginners and help you with various fundamentals of Java.


Core Java Cheat Sheet

Java is an open source programming language that has been changing the face of the IT market since ages. It is widely preferred by the programmers as the code written in Java can be executed securely on any platform, irrespective of the operating system or architecture of the device. The only requirement is, Java Runtime Environment (JRE) installed on the system.



Java Programming

Java is a high level, general purpose programming language that produces software for multiple platforms. It was developed by James Gosling in 1991 and released by Sun Microsystems in 1996 and is currently owned by Oracle.



Primitive Data Types

Type	Size	Range
byte	8	-128..127
short	16	-32,768..32,767
int	32	-2,147,483,648.. 2,147,483,647
long	64	-9,223,372,036,854,775,808.. 9,223,372,036,854,775,807
float	32	3.4e-038.. 3.4e+038
double	64	1.7e-308.. 1.7e+308
char	16	Complete Unicode Character Set
Boolean	1	True, False

Java Operators

Type	Operators
Arithmetic	+, -, *, /, %, ++, --
Assignment	=, +=, -=, *=, /=, %=, &=, ^=, <<=, >>=
Bitwise	~, &,
Logical	&&,
Relational	<, >, <=, >=, ==, !=
Shift	<<, >>, >>>
Ternary	?:
Unary	++, --, ~, !, ++x, --x, x++, x--

Java Variables

```
{public|private} [static] type name [= expression|value];
```

Java Methods

```
{public|private} [static] [type | void] name(arg1, ..., argN ){statements}
```

Data Type Conversion

```
// Widening (byte<short<int<long<float<double)
int i = 10; //int-> long
long l = i; //automatic type conversion
// Narrowing
double d = 10.02;
long l = (long)d; //explicit type casting
// Numeric values to String
String str = String.valueOf(value);
// String to Numeric values
int i = Integer.parseInt(str);
double d = Double.parseDouble(str);
```

User Input

```
// Using BufferedReader
BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));
String name = reader.readLine();
// Using Scanner
Scanner in = new Scanner(System.in);
String s = in.nextLine();
int a = in.nextInt();
// Using Console
String name = System.console().readLine();
```

Iterative Statements

```
// for loop
for (condition) (expression)

// for each loop
for (int i: someArray) {}

// while loop
while (condition) {expression}

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

Fibonacci series

```
for (i = 1; i <= n; ++i)
{
    System.out.print(t1 + " ");
    int sum = t1 + t2; t1 = t2;
    t2 = sum;
}
```

Pyramid Pattern

```
k = 2*n - 2;
for(i=0; i<n; i++)
{
    for(j=0; j<k; j++){System.out.print(" ");}
    k = k - 1;
    for(j=0; j<i; j++) {System.out.print("* ");}
    System.out.println();
}
```

Decisive Statements

```
//if statement
if (condition) (expression)

//if-else statement
if (condition) (expression) else (expression)

//switch statement
switch (var) { case 1: expression; break;
default: expression; break; }
```

Prime Number

```
if (n < 2)
{
    return false;
}
for (int i=2; i <= n/i; i++)
{
    if (n%i == 0) return false;
}
return true;
```

Factorial of a Number

```
int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return(n * factorial(n-1));
}
```

Arrays In Java

1 - Dimensional

```
// Initializing
type[] varName= new type[size];

// Declaring
type[] varName= new type[(values1, value2,...)];

Array with Random Variables
double[] arr = new double[n];
for (int i=0; i<n; i++)
{a[i] = Math.random();}
```

Maximum value in an Array

```
double max = 0;
for (int i=0; i<arr.length(); i++)
{ if(a[i] > max) max = a[i]; }
```

Reversing an Array

```
for(int i=0; i<(arr.length())/2; i++)
{ double temp = a[i];
  a[i] = a[n-1-i];
  a[n-1-i] = temp; }
```

Multi - Dimensional Arrays

```
// Initializing
datatype[][] varName = new datatype[row][col];
// Declaring
datatype[][] varName = {(value1, value2,...),(value1, value2,...)-...};

Transposing A Matrix
for(i = 0; i < row; i++)
{ for(j = 0; j < column; j++)
{ System.out.print(array[i][j]+" "); }
  System.out.println(" ");
}

Multiplying two Matrices
for (i = 0; i < row1; i++)
{ for (j = 0; j < col2; j++)
{ for (k = 0; k < row2; k++)
{ sum = sum + first[i][k]*second[k][j]; }
  multiply[i][j] = sum;
sum = 0; } } }
```

Java Strings

```
// Creating String using literal
String str1 = "Welcome";

// Creating String using new keyword
String str2 = new String("Edureka");

String Methods
str1=str2 //compare the address;
String newStr = str1.equals(str2); //compares the values
String newStr = str1.equalsIgnoreCase() //
newStr = str1.length() //calculates length
newStr = str1.charAt(i) //extract i'th character
newStr = str1.toUpperCase() //returns string in ALL CAPS
newStr = str1.toLowerCase() //returns string in ALL LOWERCASE
newStr = str1.replace(oldVal, newVal) //search and replace
newStr = str1.trim() //trims surrounding whitespace
newStr = str1.contains("value"); //Check for the values
newStr = str1.toCharArray(); //Convert into character array
newStr = str1.isEmpty(); //Check for empty String
newStr = str1.endsWith(); //Checks if string ends with the given suffix
```

Basic Java Program

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

Save


className.java

Compile

javac className

Execute

java className



Primitive Data Types

Let's start off by learning the primitive **data types** that Java offers:

Data Type	Size	Range
byte	8	-128..127
short	16	-32,768..32,767
int	32	-2,147,483,648.. 2,147,483,647
long	64	-9,223,372,036,854,775,808.. 9,223,372,036,854,775,807
float	32	3.4e-038.. 3.4e+038
double	64	1.7e-308.. 1.7e+308

Become a Certified Professional➔

Subscribe to our Newsletter, and get personalized recommendations.

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

Logical	&&,
Relational	<, >, <=, >=, ==, !=
Shift	<<, >>, >>>
Ternary	?:
Unary	++X, -X, X++, X-, +X, -X, !, ~

Java Variables

Variables in Java refers to the name of the reserved memory area. You need variables to store any value for the computational or reference purpose.

There are 3 types of variable in Java:

1. Local Variables
2. Instance Variables
3. Static Variables

```
{public | private} [static] type name [= expression | value];
```

Java Methods

A method is a set of code that is grouped together to perform a specific operation. A method is completed in two steps:

1. Method Initialization
2. Method Invocation

A method can be invoked either by calling it by reference or by value.

```
{public | private} [static] {type | void} name(arg1, ..., argN ){statements}
```

Data Conversion

The process of changing a value from one data type to another type is known as data type conversion. Data Type conversion is of two types:

1. *Widening*: The lower size datatype is converted into a higher size data type without loss of information.
2. *Narrowing*: The higher size datatype is converted into a lower size data type with a loss of information.

```
// Widening (byte<short<int<long<float<double)
int i = 10; //int--> long
long l = i; //automatic type conversion
// Narrowing
double d = 10.02;
long l = (long)d; //explicit type casting
// Numeric values to String
String str = String.valueOf(value);
// String to Numeric values
int i = Integer.parseInt(str);
double d = Double.parseDouble(str);
```

User Input

Java provides three ways to take an input from the user/ console:

1. Using BufferedReader class
2. Using Scanner class
3. Using Console class

```
// Using BufferedReader
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
String name = reader.readLine();

// Using Scanner
Scanner in = new Scanner(System.in);
String s = in.nextLine();
int a = in.nextInt();

// Using Console
String name = System.console().readLine();
```

Basic Java Program

A basic program in Java will consist of at least the following components:

1. Classes & Objects
2. Methods
3. Variables



Core Java Cheat Sheet – Basics Of Java Programming

edureka.co



Whatsapp



Linkedin



Twitter



Facebook



Reddit

Copy Link



Become a Certified Professional➔

Subscribe to our Newsletter, and get personalized recommendations.

Already have an account? [Sign in.](#)

Finally, execute the program using below code:

```
java className
```

Flow Of Control

Become a Certified Professional➔

Subscribe to our Newsletter, and get personalized recommendations.

Already have an account? [Sign in](#).



Whatsapp



Linkedin



Twitter



Facebook



Reddit

Become a Certified Professional➔

Subscribe to our Newsletter, and get personalized recommendations.

Already have an account? [Sign in.](#)



Portal Development and Text Searching with Hibernate

Watch Now




Introduction to JAVA/J2EE & SOA

Watch Now



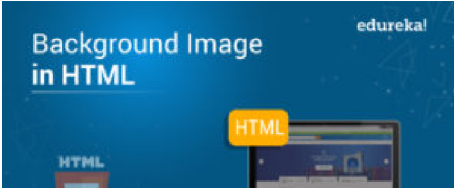
Learn Perl-the Jewel of Scripting

Watch Now



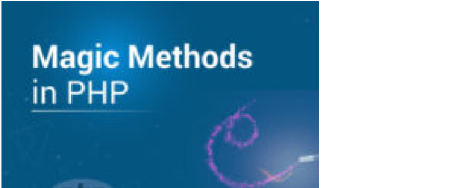
A Complete Introduction to Abstract Classes in Java

Read Article



How to Implement a Background Image in HTML

Read Article



What are magic methods in PHP? to Implement them?

Read Article



[Full Stack Web Development Internship Program](#)

👤 29k Enrolled Learners

📅 [Weekend/Weekday](#)

🖥️ [Live Class](#)

[Reviews](#)

★★★★★ 5 (11250)



[Java Certification Training Course](#)

👤 75k Enrolled Learners

📅 [Weekend](#)

🖥️ [Live Class](#)

[Reviews](#)

★★★★★ 4 (29950)



[Python Scripting Certification Training](#)

👤 14k Enrolled Learners

📅 [Weekend](#)

🖥️ [Self Paced](#)

[Reviews](#)

★★★★★ 5 (5500)



[Flutter Application Development Course](#)

👤 12k Enrolled Learners

📅 [Weekend/Weekday](#)

🖥️ [Live Class](#)

[Reviews](#)

★★★★★ 5 (4700)

Become a Certified Professional➔

Subscribe to our Newsletter, and get personalized recommendations.

Already have an account? [Sign in](#).



Whatsapp



Linkedin



Twitter



Facebook



Reddit