

Online Java Compiler - Program x Introducing ChatGPT | OpenAI x Java Programiz Compatibility x Online Java Compiler | Java Edit x

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute D

Login

1 import java.io.File;  
2 import java.io.IOException;  
3 import java.lang.management.ManagementFactory;  
4 import com.sun.management.OperatingSystemMXBean;  
5  
6 public class MaclaurinSinMetrics {  
7  
8 public static double factorial(int n) {  
9 double fact = 1;  
10 for (int i = 2; i <= n; i++) {  
11 fact \*= i;  
12 }  
13 return fact;  
14 }  
15  
16 public static double maclaurinSin(double x, int n) {  
17 double result = 0;  
18 for (int i = 0; i < n; i++) {  
19 int power = 2 \* i + 1;  
20 double term = Math.pow(-1, i) \* Math.pow(x, power) / factorial(power);  
21 result += term;  
22 }  
23 return result;  
24 }  
25  
26 public static void main(String[] args) throws IOException, InterruptedException {  
27 // === Compilation Time Measurement ===  
28 long compileStart = System.nanoTime();  
29 ProcessBuilder pb = new ProcessBuilder("javac", "MaclaurinSinMetrics.java");  
30 Process process = pb.start();  
31 process.waitFor();  
32 long compileEnd = System.nanoTime();  
33  
34 // === Memory Measurement Before Execution ===  
35 Runtime runtime = Runtime.getRuntime();  
36 runtime.gc();  
37 long usedMemoryBefore = runtime.totalMemory() - runtime.freeMemory();  
38  
39 // === Execution Time Measurement ===  
40 long startTime = System.nanoTime();  
41 double x = Math.toRadians(30);  
42 }  
43 }

Input/Output

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0 Interactive Mode

Input arguments

Output Generated files

Maclaurin approximation of sin(30°): 0.49999999999999994  
Actual value: 0.49999999999999994  
===== METRICS =====  
Compilation Time: 1181.679 ms  
Execution Time: 0.178 ms  
Memory Used: 1086664 bytes  
Code Size (.class file): 2918 bytes  
CPU Usage: 0.00%

Compiled and executed in 3.327 sec(s)

This website uses cookies to ensure you get the best experience on our website. Got it!

26°C Mostly cloudy

Search

ENG IN

11:05 09-08-2025

Online Java Compiler - Program: x | Introducing ChatGPT | OpenAI | Java Programiz Compatibility | Online Java Compiler | Java Edit | x

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute

Express

Login

26 public static void main(String[] args) throws IOException, InterruptedException {

27 // === Compilation Time Measurement ===

28 long compileStart = System.nanoTime();

29 ProcessBuilder pb = new ProcessBuilder("javac", "MaclaurinSinMetrics.java");

30 Process process = pb.start();

31 process.waitFor();

32 long compileEnd = System.nanoTime();

33

34 // === Memory Measurement Before Execution ===

35 Runtime runtime = Runtime.getRuntime();

36 runtime.gc();

37 long usedMemoryBefore = runtime.totalMemory() - runtime.freeMemory();

38

39 // === Execution Time Measurement ===

40 long startTime = System.nanoTime();

41 double x = Math.toRadians(30);

42 int terms = 10;

43 double approx = maclaurinSin(x, terms);

44 long endTime = System.nanoTime();

45

46 // === Memory Measurement After Execution ===

47 long usedMemoryAfter = runtime.totalMemory() - runtime.freeMemory();

48

49 // === CPU Usage Measurement ===

50 OperatingSystemMXBean osBean =

51 (OperatingSystemMXBean) ManagementFactory.getOperatingSystemMXBean();

52 double cpuLoad = osBean.getProcessCpuLoad() \* 100;

53

54 // === File Size Measurement ===

55 File file = new File("MaclaurinSinMetrics.class");

56

57 // === Output Results ===

58 System.out.println("Maclaurin approximation of sin(30°): " + approx);

59 System.out.println("Actual value: " + Math.sin(x));

60

61 System.out.println("\n===== METRICS =====");

62 System.out.printf("Compilation Time: %.3f ms\n", (compileEnd - compileStart) / 1\_000\_000.0);

63 System.out.printf("Execution Time: %.3f ms\n", (endTime - startTime) / 1\_000\_000.0);

64 System.out.println("Memory Used: " + (usedMemoryAfter - usedMemoryBefore) + " bytes");

65 if (file.exists()) {

66 System.out.println("Code Size (.class file): " + file.length() + " bytes");

67 }

Input/Output

API

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0

Interactive Mode

Input arguments

Output

Generated files

Maclaurin approximation of sin(30°): 0.49999999999999994

Actual value: 0.49999999999999994

===== METRICS =====

Compilation Time: 1181.679 ms

Execution Time: 0.178 ms

Memory Used: 1086664 bytes

Code Size (.class file): 2918 bytes

CPU Usage: 0.00%

Compiled and executed in 3.327 sec(s)

This website uses cookies to ensure you get the best experience on our website. Got it!

26°C Mostly cloudy

Search

ENG IN

11:05 09-08-2025

Online Java Compiler - Program x Introducing ChatGPT | OpenAI x Java Programiz Compatibility x Online Java Compiler | Java Edit x

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute

Ada Ada

Login

New Project

Online Java Compiler IDE

```
1 import java.io.File;
2 import java.io.IOException;
3 import java.lang.management.ManagementFactory;
4 import com.sun.management.OperatingSystemMXBean;
5 import java.util.Scanner;
6
7 public class PalindromeCheckMetrics {
8
9     // === Function to check if number is palindrome ===
10    public static boolean isPalindrome(int num) {
11        int original = num;
12        int reversed = 0;
13        while (num > 0) {
14            int digit = num % 10;
15            reversed = reversed * 10 + digit;
16            num /= 10;
17        }
18        return original == reversed;
19    }
20
21    public static void main(String[] args) throws IOException, InterruptedException {
22        Scanner sc = new Scanner(System.in);
23        System.out.print("Enter a number: ");
24        int num = sc.nextInt();
25        sc.close();
26
27        // === Compilation Time Measurement ===
28        long compileStart = System.nanoTime();
29        ProcessBuilder pb = new ProcessBuilder("javac", "PalindromeCheckMetrics.java");
30        Process process = pb.start();
31        process.waitFor();
32        long compileEnd = System.nanoTime();
33
34        // === Memory Measurement Before Execution ===
35        Runtime runtime = Runtime.getRuntime();
36        runtime.gc();
37        long usedMemoryBefore = runtime.totalMemory() - runtime.freeMemory();
38
39        // === Execution Time Measurement ===
40        long startTime = System.nanoTime();
41        boolean result = isPalindrome(num);
42    }
```

Input/Output

API

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0 Interactive Mode

Input arguments

Output Generated files

Enter a number: 123

===== RESULT =====

123 is NOT a Palindrome number.

===== METRICS =====

Compilation Time: 882.858 ms

Execution Time: 0.010 ms

Memory Used: 0 bytes

Code Size (.class file): 2935 bytes

CPU Usage: 0.00%

Compiled and executed in 7.256 sec(s)

This website uses cookies to ensure you get the best experience on our website. Got it!

26°C Mostly cloudy

Search

ENG IN

11:08 09-08-2025

Online Java Compiler - Program x | Introducing ChatGPT | OpenAI x | Java Programiz Compatibility x | Online Java Compiler | Java Edit x +

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute Assembler Login

```
28 long compileStart = System.nanoTime();
29 ProcessBuilder pb = new ProcessBuilder("javac", "PalindromeCheckMetrics.java");
30 Process process = pb.start();
31 process.waitFor();
32 long compileEnd = System.nanoTime();
33
34 // === Memory Measurement Before Execution ===
35 Runtime runtime = Runtime.getRuntime();
36 runtime.gc();
37 long usedMemoryBefore = runtime.totalMemory() - runtime.freeMemory();
38
39 // === Execution Time Measurement ===
40 long startTime = System.nanoTime();
41 boolean result = isPalindrome(num);
42 long endTime = System.nanoTime();
43
44 // === Memory Measurement After Execution ===
45 long usedMemoryAfter = runtime.totalMemory() - runtime.freeMemory();
46
47 // === CPU Usage Measurement ===
48 OperatingSystemMXBean osBean =
49 (OperatingSystemMXBean) ManagementFactory.getOperatingSystemMXBean();
50 double cpuLoad = osBean.getProcessCpuLoad() * 100;
51
52 // === File Size Measurement ===
53 File file = new File("PalindromeCheckMetrics.class");
54
55 // === Output Results ===
56 System.out.println("\n===== RESULT =====");
57 if (result) {
58     System.out.println(num + " is a Palindrome number.");
59 } else {
60     System.out.println(num + " is NOT a Palindrome number.");
61 }
62
63 System.out.println("\n===== METRICS =====");
64 System.out.printf("Compilation Time: %.3f ms\n", (compileEnd - compileStart) / 1_000_000.0);
65 System.out.printf("Execution Time: %.3f ms\n", (endTime - startTime) / 1_000_000.0);
66 System.out.println("Memory Used: " + (usedMemoryAfter - usedMemoryBefore) + " bytes");
67 if (file.exists()) {
68     System.out.println("Code Size (.class file): " + file.length() + " bytes");
69 }
```

Input/Output API

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0 Interactive Mode

Input arguments

Output Generated files

Enter a number: 123  
===== RESULT =====  
123 is NOT a Palindrome number.  
  
===== METRICS =====  
Compilation Time: 882.858 ms  
Execution Time: 0.010 ms  
Memory Used: 0 bytes  
Code Size (.class file): 2935 bytes  
CPU Usage: 0.00%

Compiled and executed in 7.256 sec(s)

This website uses cookies to ensure you get the best experience on our website. Got it!

Gold +0.84%

Search

ENG IN

11:08 09-08-2025

Online Java Compiler - Program... x Introducing ChatGPT | OpenAI x Java Programiz Compatibility x Online Java Compiler | Java Edit x

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute Unlambda Login

```
1 import java.io.File;
2 import java.io.IOException;
3 import java.lang.management.ManagementFactory;
4 import java.lang.reflect.Method;
5 import java.util.Scanner;
6
7 public class DeepRecursionMetrics {
8
9     // Simple deep recursion (counts down)
10    public static long deepRecursion(int n) {
11        if (n <= 0) return 0;
12        return 1 + deepRecursion(n - 1);
13    }
14
15    // Try to get process CPU load via reflection (safe in restricted JREs)
16    private static Double getProcessCpuLoadPercent() {
17        try {
18            Object osBean = ManagementFactory.getOperatingSystemMXBean();
19            // Use reflection so code compiles even if com.sun.* classes are absent
20            Class<?> sunClass = Class.forName("com.sun.management.OperatingSystemMXBean");
21            if (sunClass.isInstance(osBean)) {
22                Method m = sunClass.getMethod("getProcessCpuLoad");
23                Object val = m.invoke(osBean);
24                if (val instanceof Double) {
25                    double d = (Double) val;
26                    return (d >= 0.0) ? d * 100.0 : null;
27                }
28            }
29        } catch (ClassNotFoundException e) {
30            // com.sun.management not available in this environment
31        } catch (Exception e) {
32            // method not accessible / invocation error - ignore and return null
33        }
34        return null;
35    }
36
37    public static void main(String[] args) {
38        Scanner sc = new Scanner(System.in);
39        System.out.print("Enter depth of recursion (try small values first, e.g. 1000): ");
40        int depth = sc.nextInt();
41        sc.close();
42    }
43 }
```

Input/Output API

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0 Interactive Mode

Input arguments

Output Generated files

Enter depth of recursion (try small values first, e.g. 1000): 1001  
Recursion depth reached (count): 1001

===== METRICS =====  
Compilation Time: 794.774 ms  
Execution Time: 7.766 ms  
Memory Used (approx): 1510016 bytes  
Code Size (.class file): 4542 bytes  
CPU Usage (process): 0.00%

Compiled and executed in 9.788 sec(s)

This website uses cookies to ensure you get the best experience on our website.

Got it!

Online Java Compiler - Program x | Introducing ChatGPT | OpenAI x | Java Programiz Compatibility x | Online Java Compiler | Java Edit x +

jdoodle.com/online-java-compiler

Online Java Compiler IDE

Execute ▶ Ada Ada

Login

43 // ----- try to measure compilation time (best-effort; may fail in JDoodle) -----  
44 long compileStart = 0L, compileEnd = 0L;  
45 boolean compileMeasured = false;  
46 try {  
47 compileStart = System.nanoTime();  
48 ProcessBuilder pb = new ProcessBuilder("javac", "DeepRecursionMetrics.java");  
49 pb.redirectErrorStream(true);  
50 Process p = pb.start();  
51 int exit = p.waitFor(); // may throw or return non-zero in restricted environments  
52 compileEnd = System.nanoTime();  
53 compileMeasured = true;  
54 if (exit != 0) {  
55 System.out.println("Note: javac returned exit code " + exit + " while measuring compilation.");  
56 }  
57 } catch (IOException | InterruptedException | SecurityException ex) {  
58 System.out.println("Compilation timing not available in this environment: " + ex.getClass().getSimpleName());  
59 // continue - compileMeasured remains false  
60 }  
61  
62 // ----- Memory before -----  
63 Runtime runtime = Runtime.getRuntime();  
64 runtime.gc();  
65 long memBefore = runtime.totalMemory() - runtime.freeMemory();  
66  
67 // ----- Execution (with StackOverflow handling) -----  
68 long startTime = System.nanoTime();  
69 long result = -1;  
70 boolean stackOverflow = false;  
71 try {  
72 result = deepRecursion(depth);  
73 System.out.println("Recursion depth reached (count): " + result);  
74 } catch (StackOverflowError so) {  
75 stackOverflow = true;  
76 System.out.println("ERROR: StackOverflowError - recursion too deep for JVM stack.");  
77 }  
78 long endTime = System.nanoTime();  
79  
80 // ----- Memory after -----  
81 long memAfter = runtime.totalMemory() - runtime.freeMemory();  
82  
83 // ----- CPU usage (best-effort via reflection) -----  
84 Double cpuPercent = getProcessCpuLoadPercent();

Input/Output

Build beautiful web apps with JDoodle.ai

Language version: JDK 21.0.0 ☐ Interactive Mode

Input arguments

Output Generated files

Enter depth of recursion (try small values first, e.g. 1000): 1001  
Recursion depth reached (count): 1001  
  
===== METRICS =====  
Compilation Time: 794.774 ms  
Execution Time: 7.766 ms  
Memory Used (approx): 1510016 bytes  
Code Size (.class file): 4542 bytes  
CPU Usage (process): 0.00%  
  
Compiled and executed in 9.788 sec(s)

This website uses cookies to ensure you get the best experience on our website.

Got it!

26°C  
Mostly cloudy

Search

ENG  
IN

11:16  
09-08-2025

