



```
1 package basic;
2
3 public class input2 {
4
5     public static void main(String[] args) {
6         int x = 7;
7
8         long y = x;
9
10        float z = y;
11        System.out.println("Before conversion, int value "+x);
12        System.out.println("After conversion, long value "+y);
13        System.out.println("After conversion, float value "+z);
14    }
15
16
17 }
18
19
```

basic

- input2
 - main(String[]): void

Problems Javadoc Declaration Console ×

<terminated> input2 [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.e

Before conversion, int value 7
After conversion, long value 7
After conversion, float value 7.0

Writable Smart Insert 9:9:125

JAVA PROGRAM FOR IMPLICIT AND EXPLICIT TYPE CASTING

JAVA PROGRAMM FOR ACCESS MODIFIER

```
1 class Data {  
2     // private variable  
3     String name;  
4 }  
5  
6 public class Main {  
7     public static void main(String[] main){  
8  
9  
10        Data d = new Data();  
11  
12  
13        d.name = "Programiz";  
14    }  
15 }
```



Problems @ Javadoc Declaration Console ×
<terminated> Main [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe

Writable

Smart Insert

3 : 5 : 43

JAVA PROGRAMM FOR COLECTIONS

```
package CollectioExample;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;

public class CollectionExample {

    public static void main(String[] args) {

        ArrayList<String> list = new ArrayList<String>();

        list.add("apple");
        list.add("banana");
        list.add("cherry");
        list.add("date");

        System.out.println("ArrayList:");
        for (String s : list) {
            System.out.println(s);
        }

        HashSet<Integer> set = new HashSet<Integer>();

        set.add(10);
        set.add(20);
        set.add(30);
        set.add(40);

        System.out.println("\nHashSet:");
        for (int i : set) {
            System.out.println(i);
        }

        HashMap<String, Integer> map = new HashMap<String, Integer>();

        map.put("one", 1);
        map.put("two", 2);
        map.put("three", 3);
        map.put("four", 4);
```

```
System.out.println("\nHashMap:");
for (String key : map.keySet()) {
    int value = map.get(key);
    System.out.println(key + ": " + value);
}
}
```

```
HashMap:
four: 4
one: 1
two: 2
three: 3
```

JAVA PROGRAMM FOR MAPS

```
package MapExample;

import java.util.HashMap;
import java.util.Map;

public class MapExample {

    public static void main(String[] args) {

        Map<String, Integer> map = new HashMap<String, Integer>();

        map.put("apple", 1);
        map.put("banana", 2);
        map.put("cherry", 3);
        map.put("date", 4);

        System.out.println("Original map:");
        for (String key : map.keySet()) {
            int value = map.get(key);
            System.out.println(key + ": " + value);
        }

        map.put("banana", 5);

        System.out.println("\nModified map:");
```

```
for (String key : map.keySet()) {  
    int value = map.get(key);  
    System.out.println(key + ": " + value);  
}
```

```
map.remove("cherry");
```

```
System.out.println("\nFinal map:");  
for (String key : map.keySet()) {  
    int value = map.get(key);  
    System.out.println(key + ": " + value);  
}  
}
```

```
}
```

Original map:

banana: 2
date: 4
apple: 1
cherry: 3

Modified map:

banana: 5
date: 4
apple: 1
cherry: 3

Final map:

banana: 5
date: 4
apple: 1

JAVA PROGRAMM FOR INNER CLASS

```
1 package InnerClassExample;
2 public class InnerClassExample {
3
4     private int outerValue;
5
6     public InnerClassExample(int value) {
7         outerValue = value;
8     }
9
10    public void printValues() {
11        System.out.println("Outer value: " + outerValue);
12        InnerClass inner = new InnerClass();
13        inner.printValue();
14    }
15
16    public class InnerClass {
17
18        private int innerValue;
19
20        public InnerClass() {
21            innerValue = outerValue + 1;
22        }
23
24        public void printValue() {
25            System.out.println("Inner value: " + innerValue);
26        }
27    }
28
29    public static void main(String[] args) {
30        InnerClassExample outer = new InnerClassExample(10);
31        outer.printValues();
32    }
```

InnerClassE
> InnerClassE

Problems Javadoc Declaration Console ×

<terminated> InnerClassExample [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.
Outer value: 10
Inner value: 11

JAVA PROGRAMM FOR REGULAR EXPRESSION

```
1 package RegexExample;
2 import java.util.regex.Matcher;
3
4
5 public class RegexExample {
6
7     public static void main(String[] args) {
8
9         Pattern pattern = Pattern.compile("\\bcat\\b");
10
11
12         String text = "The cat sat on the mat.";
13
14
15         Matcher matcher = pattern.matcher(text);
16         while (matcher.find()) {
17             System.out.println("Match found at position " + matcher.start());
18         }
19     }
20
21 }
22
23
```

Problems @ Javadoc Declaration Console ×

<terminated> RegexExample [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.xi
Match found at position 4

JAVA PROGRAM SRTING CONVERSION

```
1 package StringConversionExample.java;
2 public class StringConversionExample {
3
4     public static void main(String[] args) {
5         String str = "Hello, world!";
6         System.out.println("Original string: " + str);
7
8         StringBuffer buffer = new StringBuffer(str);
9         System.out.println("String converted to StringBuffer: " + buffer);
10
11         StringBuilder builder = new StringBuilder(str);
12         System.out.println("String converted to StringBuilder: " + builder);
13     }
14 }
15 }
16
```

StringConversionEx
StringConversionEx
main(String[]) :

Problems @ Javadoc Declaration Console ×

terminated> StringConversionExample [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7

Original string: Hello, world!
string converted to StringBuffer: Hello, world!
string converted to StringBuilder: Hello, world!

JAVA PROGRAMM FOR IMPLEMENTATION OF ARRAYS

The screenshot displays the Eclipse IDE with a Java project named 'GFG'. The 'Arrays.java' file is open, showing the following code:

```
1 package GFG;
2
3 public class Arrays {
4
5     public static void main(String[] args) {
6         int[] arr;
7
8
9         arr = new int[5];
10
11
12         arr[0] = 10;
13
14
15         arr[1] = 20;
16
17
18         arr[2] = 30;
19         arr[3] = 40;
20         arr[4] = 50;
21
22
23         for (int i = 0; i < arr.length; i++)
24             System.out.println("Element at index " + i
25                               + " : " + arr[i]);
26     }
27 }
28
29
30
31
```

The Outline view on the right shows the project structure with 'GFG' and 'Arrays' (containing 'main(String[]) : void'). The Console view at the bottom shows the output of the program:

```
<terminated> Arrays [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\jav
Element at index 0 : 10
Element at index 1 : 20
Element at index 2 : 30
Element at index 3 : 40
Element at index 4 : 50
```

JAVA PROGRAMM FOR CONSTRUCTOR TYPES

```
package myclass1;
public class myclass1 {

    private int x;
    private int y;

    public myclass1() {
        x = 0;
        y = 0;
    }

    public myclass1(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    public static void main(String[] args) {

        myclass1 obj1 = new myclass1();
        System.out.println("obj1 x: " + obj1.getX());
        System.out.println("obj1 y: " + obj1.getY());

        myclass1 obj2 = new myclass1(5, 10);
        System.out.println("obj2 x: " + obj2.getX()); // prints "5"
        System.out.println("obj2 y: " + obj2.getY()); // prints "10"

        obj2.setX(20);
        obj2.setY(30);
        System.out.println("obj2 x: " + obj2.getX());
        System.out.println("obj2 y: " + obj2.getY());
    }
}
```

<terminated> myclass1 [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\jav

```
obj1 x: 0  
obj1 y: 0  
obj2 x: 5  
obj2 y: 10  
obj2 x: 20  
obj2 y: 30
```

JAVA PROGRAM FOR CALLING A FUNCTION

```
package basic;

public class MyClass {

    public static void main(String[] args) {

        MyClass myObj = new MyClass();

        myObj.printMessage("Hello, world!");

        int result = myObj.addNumbers(5, 10);
        System.out.println("The result is: " + result);

        int[] numbers = {1, 2, 3, 4, 5};
        int max = myObj.findMax(numbers);
        System.out.println("The maximum number is: " + max);
    }

    public void printMessage(String message) {
        System.out.println(message);
    }

    public int addNumbers(int num1, int num2) {
        return num1 + num2;
    }

    public int findMax(int[] nums) {
        int max = nums[0];
        for (int i = 1; i < nums.length; i++) {
            if (nums[i] > max) {
                max = nums[i];
            }
        }
        return max;
    }
}
```

```
Hello, world!
The result is: 15
The maximum number is: 5
```