

# ◆ Arrays



## ◆ Points to Remember:-

- ◆ All the elements in the array should be of same data type.
- ◆ All the elements are stored in adjacent memory location for a quicker and convenient access.

## ◆ Initialization of Array:-

- ◆ Syntax:- `<name of datatype>[] <name of variable> = {val1,val2...};`
- ◆ Example:

The screenshot shows an IDE with a Java file named `Main.java`. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String[] names = {"Study", "easy"};
7         System.out.println(names);
8     }
9
10 }
11
```

The console output at the bottom shows:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
[Ljava.lang.String;@626b2d4a
```

A blue arrow points from the memory address `@626b2d4a` in the console to the text 'Base Address of Array Element'.

Base Address of Array Element

The indices always start from zero. If you want to access array element use `indices()`.

The screenshot shows the same IDE with the `Main.java` file. The code is modified to print each element of the array using its index:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String[] names = {"Study", "easy"};
7         System.out.println(names[0]);
8         System.out.println(names[1]);
9     }
10 }
11
12
```

The console output now shows:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 3, 2024, 5:26
Study
easy
```

If you are trying to access the element which is not present, then you get an exception which is unchecked exception - `ArrayIndexOutOfBoundsException`.

The screenshot shows an IDE with a Java file named `Main.java`. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String[] names = {"Study", "easy"};
7         System.out.println(names[3]);
8     }
9
10 }
11
```

The console output shows the following error:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 3, 2024, 5:29:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2
    at org.studyeasy.Main.main(Main.java:7)
```

## Without Intializing

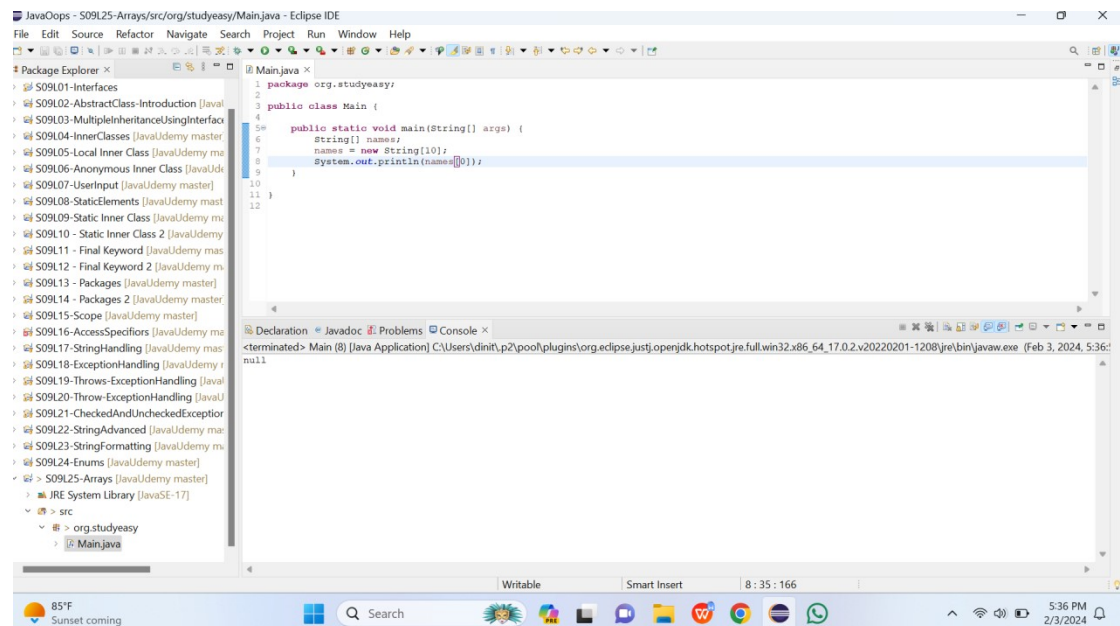
What happens if we do not mention the length of the array? Then we get below error.

The screenshot shows an IDE with a Java file named `Main.java`. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String[] names;
7         System.out.println(names);
8     }
9
10 }
11
```

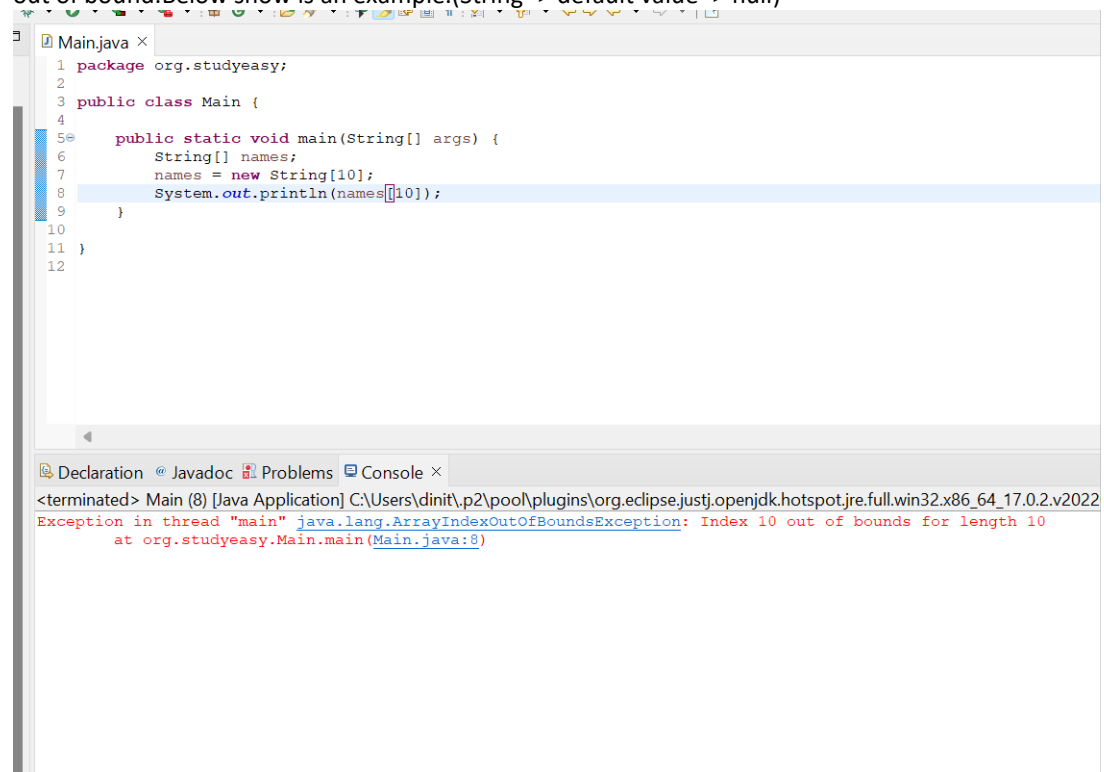
The console output shows the following error:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 3, 2024, 5:3
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The local variable names may not have been initialized
    at org.studyeasy.Main.main(Main.java:7)
```

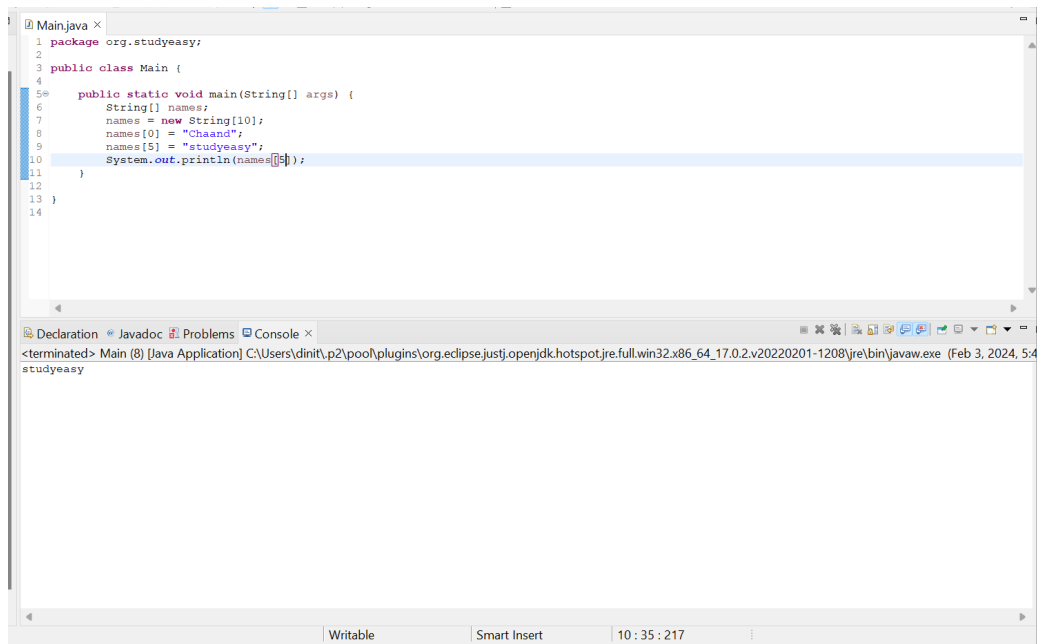


If value not assigned then, Java will assign null to all values for the specified length.

If we try to run the program where index is out of array, then we get the error stating that Index 10 out of bound. Below shown is an example. (String -> default value -> null)



Assigning the values to the array:-



```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String[] names;
7         names = new String[10];
8         names[0] = "Chaand";
9         names[5] = "studyeasy";
10        System.out.println(names[5]);
11    }
12 }
13
14
```

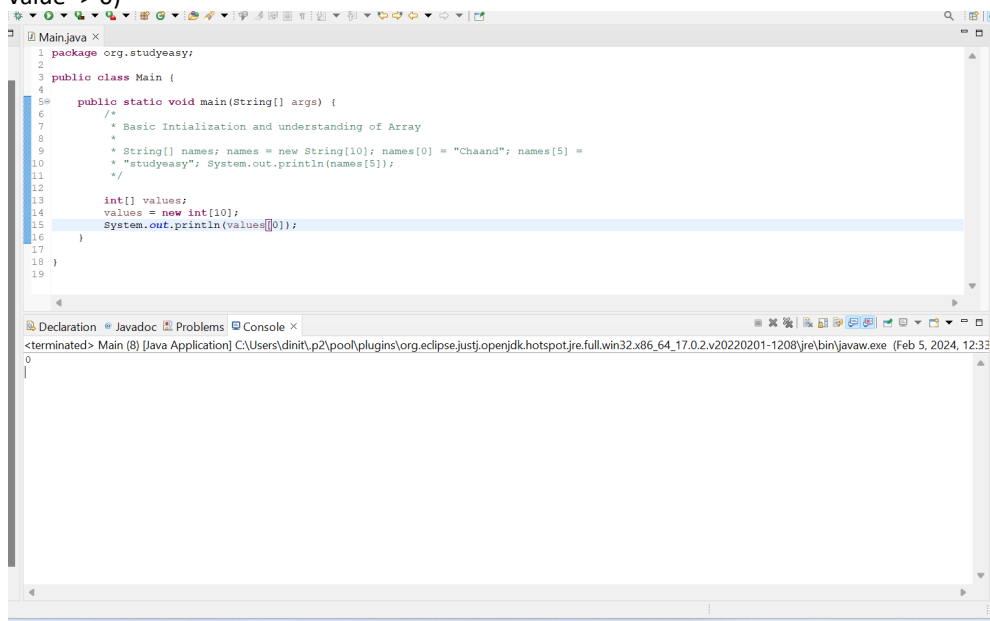
Declaration Javadoc Problems Console

<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 3, 2024, 5:4)

studyeasy

Writable Smart Insert 10 : 35 : 217

If we do not specify any value to the string array, then his what happens! (int Data Type -> default value -> 0)



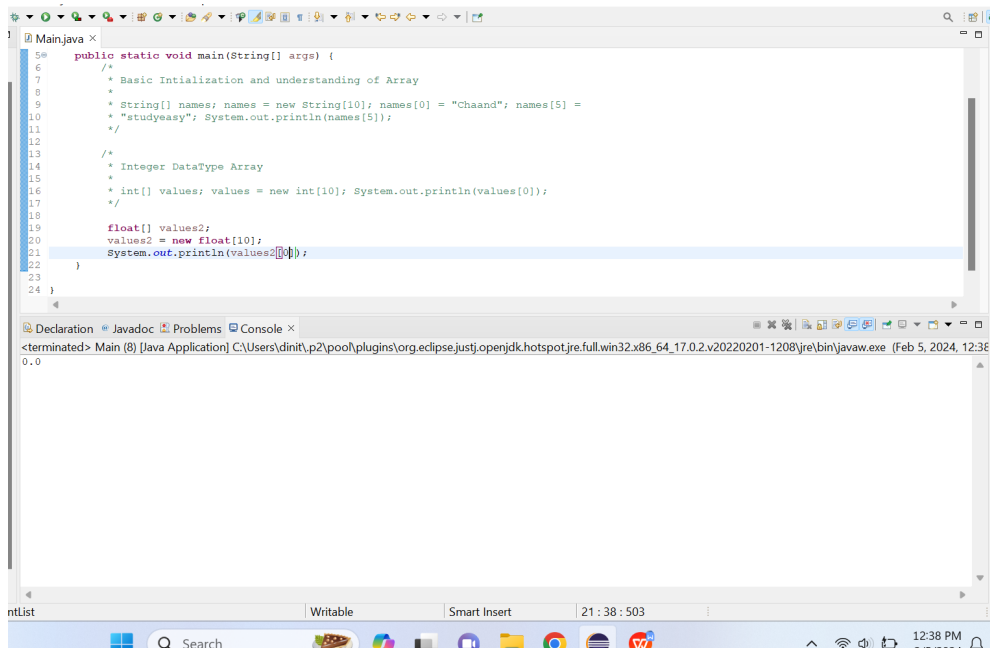
```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         /*
7          * Basic Initialization and understanding of Array
8          *
9          * String[] names; names = new String[10]; names[0] = "Chaand"; names[5] =
10         * "studyeasy"; System.out.println(names[5]);
11         */
12
13         int[] values;
14         values = new int[10];
15         System.out.println(values[0]);
16     }
17 }
18
19
```

Declaration Javadoc Problems Console

<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 5, 2024, 12:33)

0

If we do not specify any value to the string array, then his what happens! (float Data Type -> default value -> 0.0)



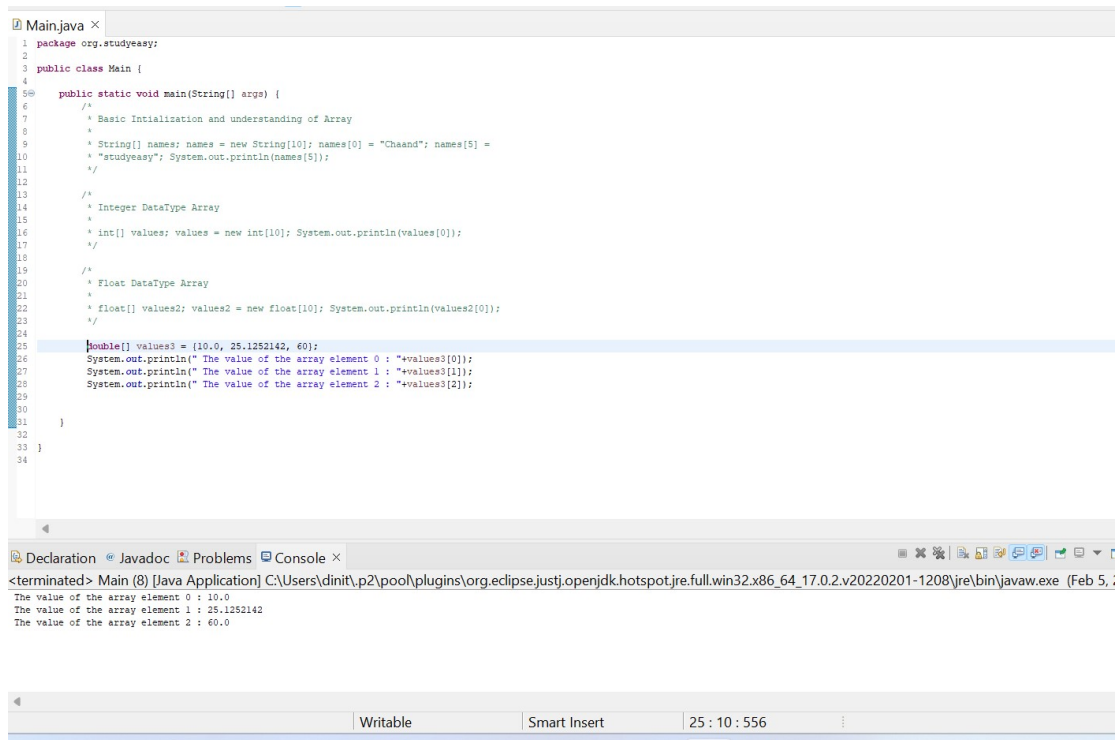
The screenshot shows an IDE window titled 'Main.java'. The code is as follows:

```
5 public static void main(String[] args) {
6     /*
7      * Basic Initialization and understanding of Array
8      */
9     * String[] names; names = new String[10]; names[0] = "Chaand"; names[5] =
10    * "studyeasy"; System.out.println(names[5]);
11    */
12
13    /*
14     * Integer DataType Array
15     */
16    * int[] values; values = new int[10]; System.out.println(values[0]);
17    */
18
19    * float[] values2;
20    values2 = new float[10];
21    System.out.println(values2[0]);
22 }
23
24 }
```

The bottom panel shows the 'Console' tab with the following output:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 5, 2024, 12:38 PM)
0.0
```

## Assigning the values to the double type



The screenshot shows an IDE window titled 'Main.java'. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         /*
7          * Basic Initialization and understanding of Array
8          */
9         * String[] names; names = new String[10]; names[0] = "Chaand"; names[5] =
10        * "studyeasy"; System.out.println(names[5]);
11        */
12
13        /*
14         * Integer DataType Array
15         */
16        * int[] values; values = new int[10]; System.out.println(values[0]);
17        */
18
19        /*
20         * Float DataType Array
21         */
22        * float[] values2; values2 = new float[10]; System.out.println(values2[0]);
23        */
24
25        double[] values3 = {10.0, 25.1252142, 60};
26        System.out.println(" The value of the array element 0 : "+values3[0]);
27        System.out.println(" The value of the array element 1 : "+values3[1]);
28        System.out.println(" The value of the array element 2 : "+values3[2]);
29
30    }
31 }
32
33 }
34 }
```

The bottom panel shows the 'Console' tab with the following output:

```
<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 5, 2024, 12:38 PM)
The value of the array element 0 : 10.0
The value of the array element 1 : 25.1252142
The value of the array element 2 : 60.0
```

```
3 public class Main {
4
5     public static void main(String[] args) {
6         /*
7          * Basic Initialization and understanding of Array
8          *
9          * String[] names; names = new String[10]; names[0] = "Chaand"; names[5] =
10          * "studyeasy"; System.out.println(names[5]);
11          */
12
13         /*
14          * Integer DataType Array
15          *
16          * int[] values; values = new int[10]; System.out.println(values[0]);
17          */
18
19         /*
20          * Float DataType Array
21          *
22          * float[] values2; values2 = new float[10]; System.out.println(values2[0]);
23          */
24
25         /*
26          * Double Array Element Recognition
27          *
28          * double[] values3 = {10.0, 25.1252142, 60};
29          * System.out.println(" The value of the array element 0 : "+values3[0]);
30          * System.out.println(" The value of the array element 1 : "+values3[1]);
31          * System.out.println(" The value of the array element 2 : "+values3[2]);
32          */
33
34
35         float[] values3 = {10.0, 25.1252142, 60};
36         System.out.println(values3[0]);
37     }
38 }
39
40
```

<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 5, 2024, 12:58)

Exception in thread "main" java.lang.Error: Unresolved compilation problems:  
Type mismatch: cannot convert from double to float  
at org.studyeasy.Main.main(Main.java:35)

What's going on here? Why there is error? Its because if we create 10.0, it will be double value. As a result we should explicitly specify the literal **f** to floating point number. Refer below screenshot.

```
18
19
20     /*
21      * Float DataType Array
22      *
23      * float[] values2; values2 = new float[10]; System.out.println(values2[0]);
24      */
25
26     /*
27      * Double Array Element Recognition
28      *
29      * double[] values3 = {10.0, 25.1252142, 60};
30      * System.out.println(" The value of the array element 0 : "+values3[0]);
31      * System.out.println(" The value of the array element 1 : "+values3[1]);
32      * System.out.println(" The value of the array element 2 : "+values3[2]);
33      */
34
35     /*
36      * Floating Point Error (Double and Float)
37      *
38      * float[] values3 = {10.0, 25.1252142, 60}; System.out.println(values3[2]);
39      */
40
41     float[] values3 = {10.0f, 25.1252142f, 60};
42     System.out.println(" The value of the array element 0 : "+values3[0]);
43     System.out.println(" The value of the array element 1 : "+values3[1]); //Here 2 at decimal point for the given literal gets
44     System.out.println(" The value of the array element 2 : "+values3[2]); //truncated
45 }
46
47
48 }
49
```

<terminated> Main (8) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.2.v20220201-1208\jre\bin\javaw.exe (Feb 5, 2024, 12:58)

The value of the array element 0 : 10.0  
The value of the array element 1 : 25.125214  
The value of the array element 2 : 60.0