



# Welcome To Tech By WebCoder

## DSA Series



# Topic : 1



## Operation In Array

1. Insert
2. Remove
3. Select
4. Update

|   |   |   |   |
|---|---|---|---|
| 2 | 5 | 1 |   |
| 0 | 1 | 2 | 3 |

**Array**



1.

**INSERT**

Operation In  
Array

#techbywebcoder

# INSERT Operation In Array



## Algorithm

Step 1: Start

Step 2: Input value to insert

Step 3: Input position where the value should be inserted

Step 4: If position < 0 OR position > size OR array is full

Print "Invalid position" and stop

Step 5: For i = size-1 down to position

Move arr[i] to arr[i + 1]

Step 6: Insert value at arr[position]

Step 7: Increase size by 1

Step 8: Stop

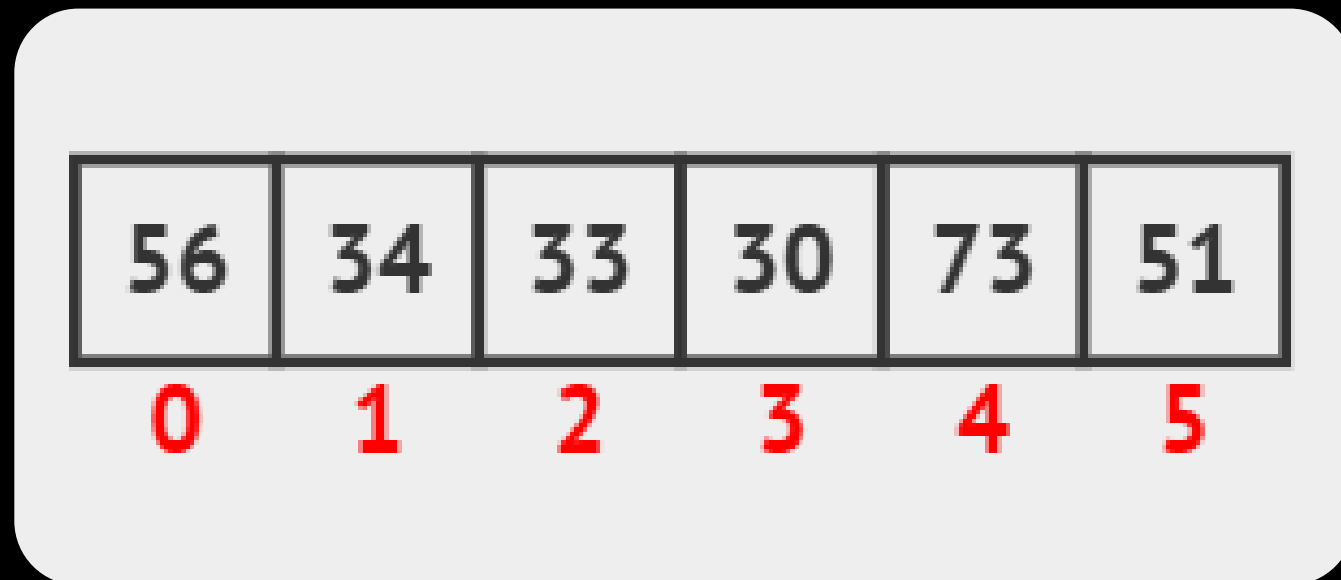
#techbywebcoder

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 56 | 34 | 33 | 30 | 73 | 51 |
| 0  | 1  | 2  | 3  | 4  | 5  |

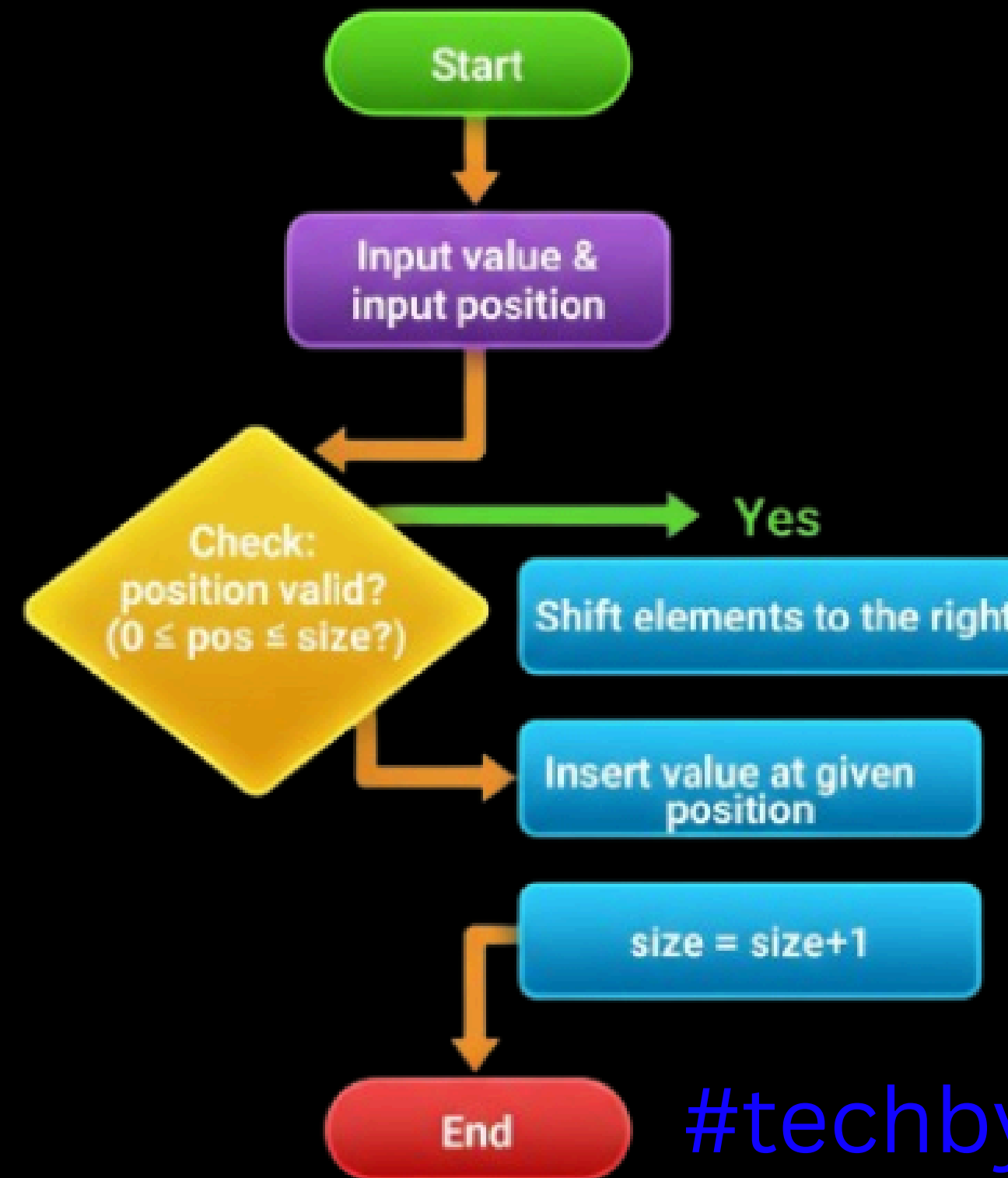
**Insert 30 to index  
3 of array**

# INSERT Operation In Array

## Flowchart

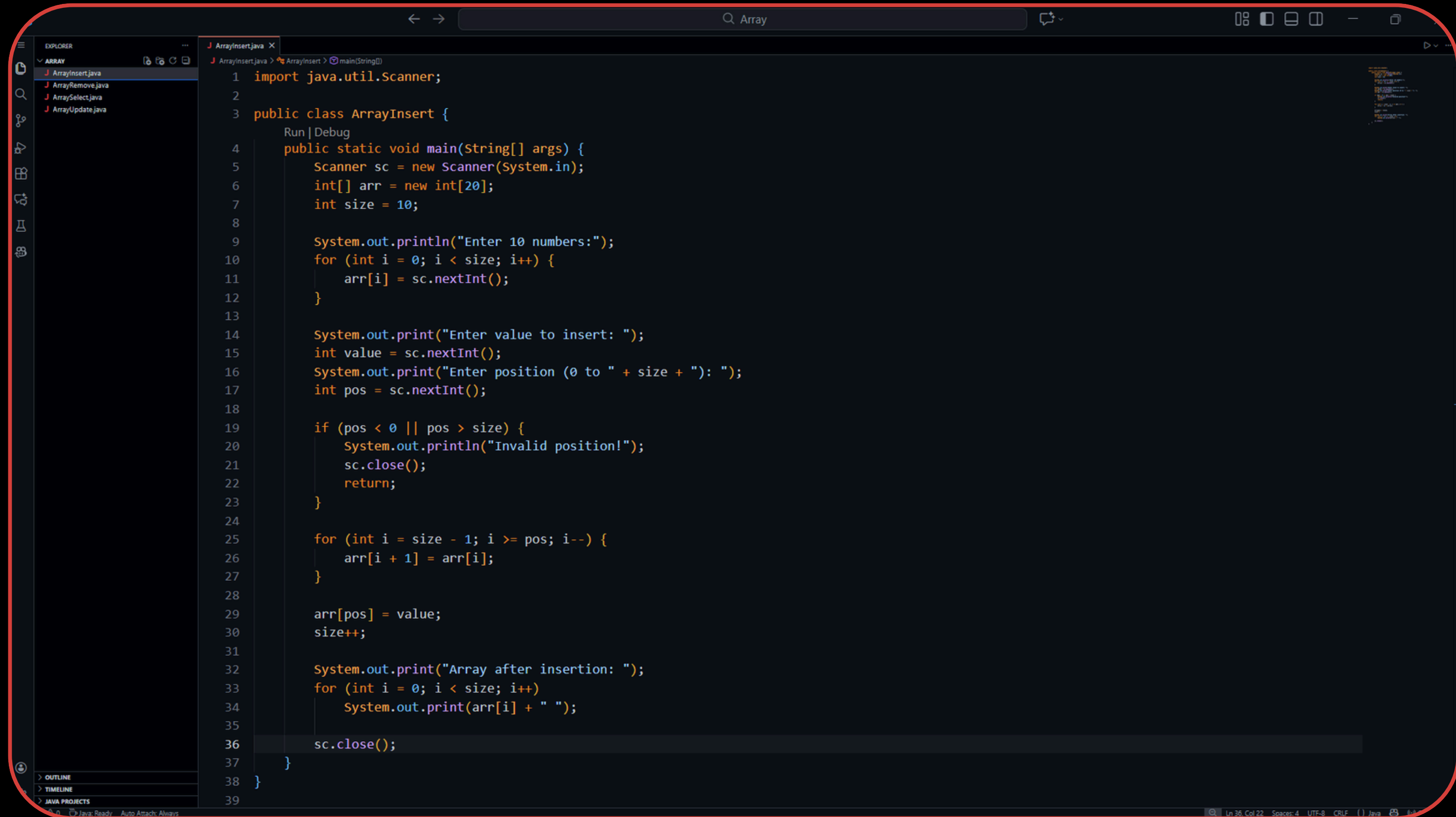


**Insert 30 to index  
3 of array**



#techbywebcoder

# Program



The screenshot shows an IDE with a dark theme. The Explorer panel on the left shows a project named 'ARRAY' with four files: 'ArrayInsert.java', 'ArrayRemove.java', 'ArraySelect.java', and 'ArrayUpdate.java'. The main editor displays the code for 'ArrayInsert.java'. The code is as follows:

```
1 import java.util.Scanner;
2
3 public class ArrayInsert {
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int[] arr = new int[20];
8         int size = 10;
9
10        System.out.println("Enter 10 numbers:");
11        for (int i = 0; i < size; i++) {
12            arr[i] = sc.nextInt();
13        }
14
15        System.out.print("Enter value to insert: ");
16        int value = sc.nextInt();
17        System.out.print("Enter position (0 to " + size + "): ");
18        int pos = sc.nextInt();
19
20        if (pos < 0 || pos > size) {
21            System.out.println("Invalid position!");
22            sc.close();
23            return;
24        }
25
26        for (int i = size - 1; i >= pos; i--) {
27            arr[i + 1] = arr[i];
28        }
29
30        arr[pos] = value;
31        size++;
32
33        System.out.print("Array after insertion: ");
34        for (int i = 0; i < size; i++)
35            System.out.print(arr[i] + " ");
36
37        sc.close();
38    }
39 }
```

The status bar at the bottom indicates 'Ln 36, Col 22, Spaces: 4, UTF-8, CRLF, Java'.

# Output

```
C:\Windows\System32\cmd.e  X  +  v

D:\system\DSA\1. Array Operation\Array>javac ArrayInsert.java

D:\system\DSA\1. Array Operation\Array>java ArrayInsert
Enter The 10 Number
10
20
30
40
50
60
70
80
90
100
Enter Value To Insert : 7
Enter Position (0 to 10):1
Array After Inseration :10
7
20
30
40
50
60
70
80
90
100

D:\system\DSA\1. Array Operation\Array>
```



# 2.

# REMOVE

# Operation In Array

#techbywebcoder



# REMOVE Operation In Array



## Algorithm

Step 1: Start

Step 2: Input position to remove

Step 3: If position  $< 0$  OR position  $\geq$  size

Print "Invalid position" and stop

Step 4: For  $i = \text{position}$  to  $\text{size}-2$

Move  $\text{arr}[i + 1]$  to  $\text{arr}[i]$

Step 5: Decrease size by 1

Step 6: Stop

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 56 | 34 | 33 | 30 | 73 | 51 |
| 0  | 1  | 2  | 3  | 4  | 5  |

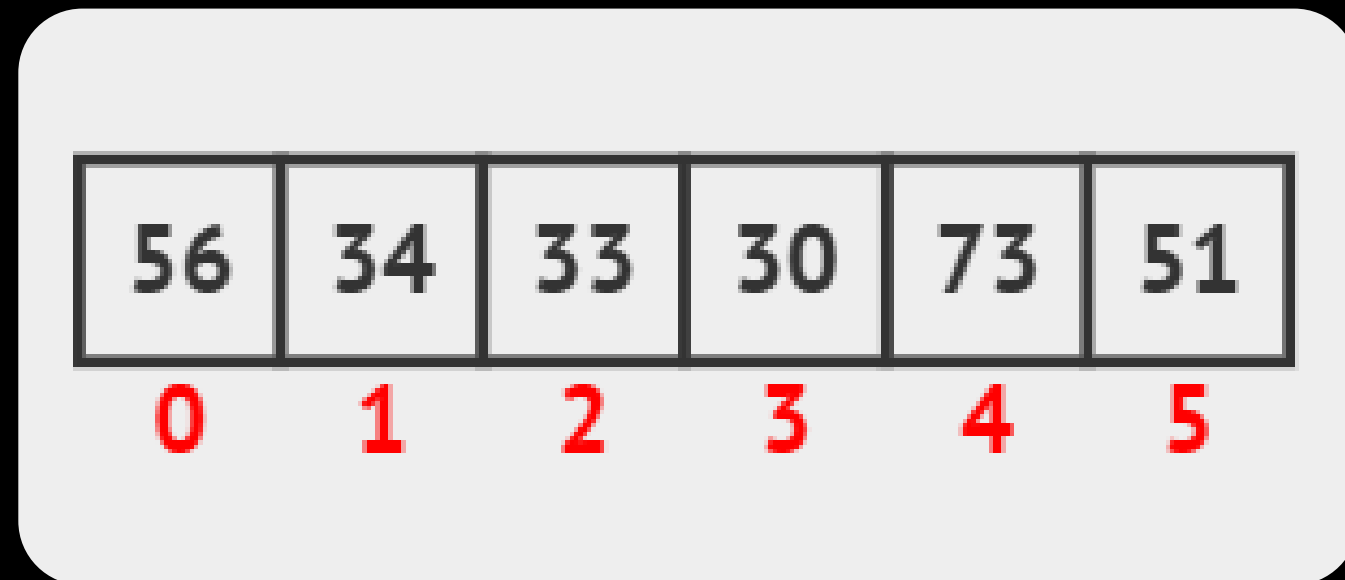
**Remove  $\text{array}[i]$   
from array**

#techbywebcoder

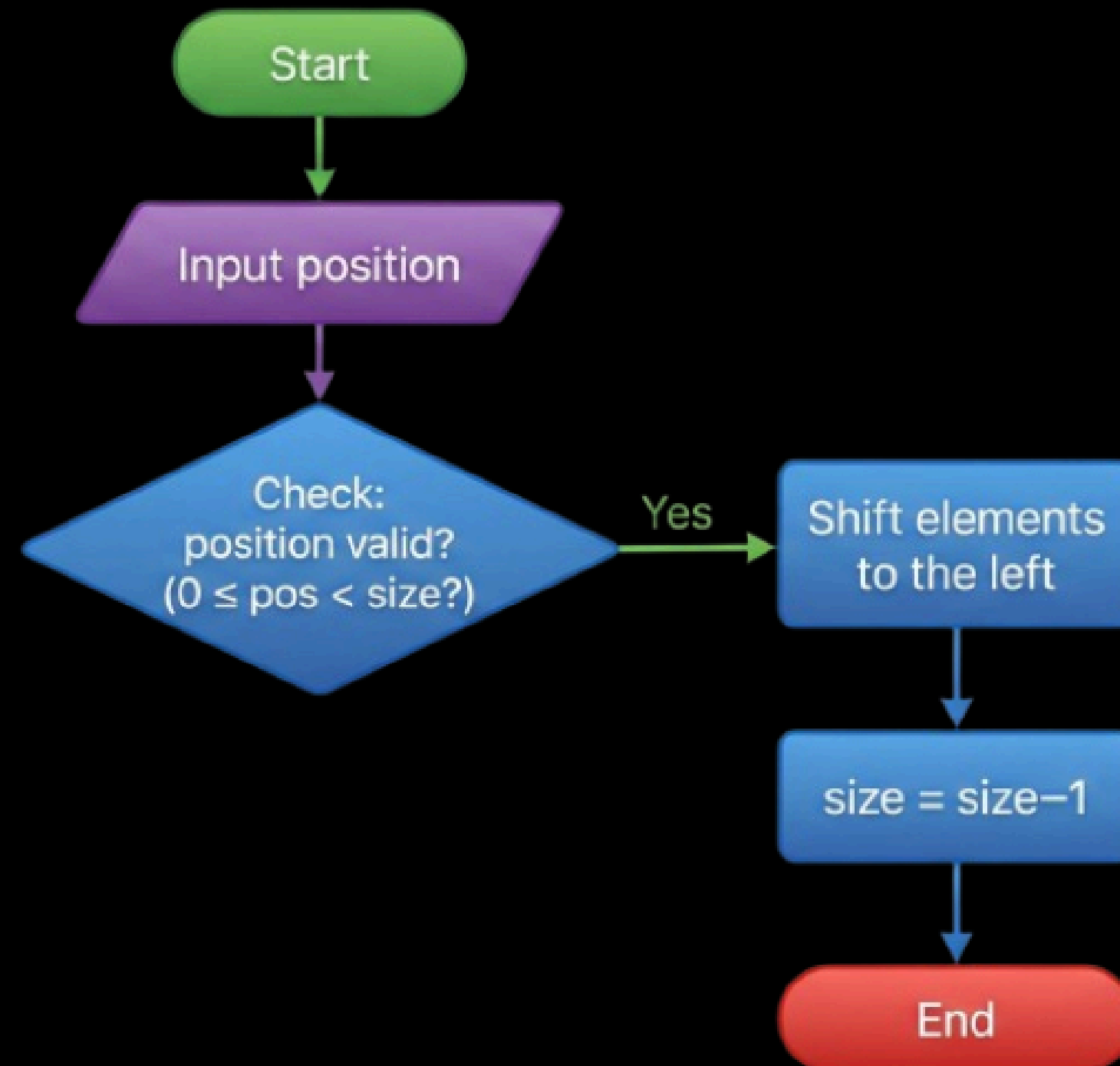
# REMOVE Operation In Array



## Flowchart

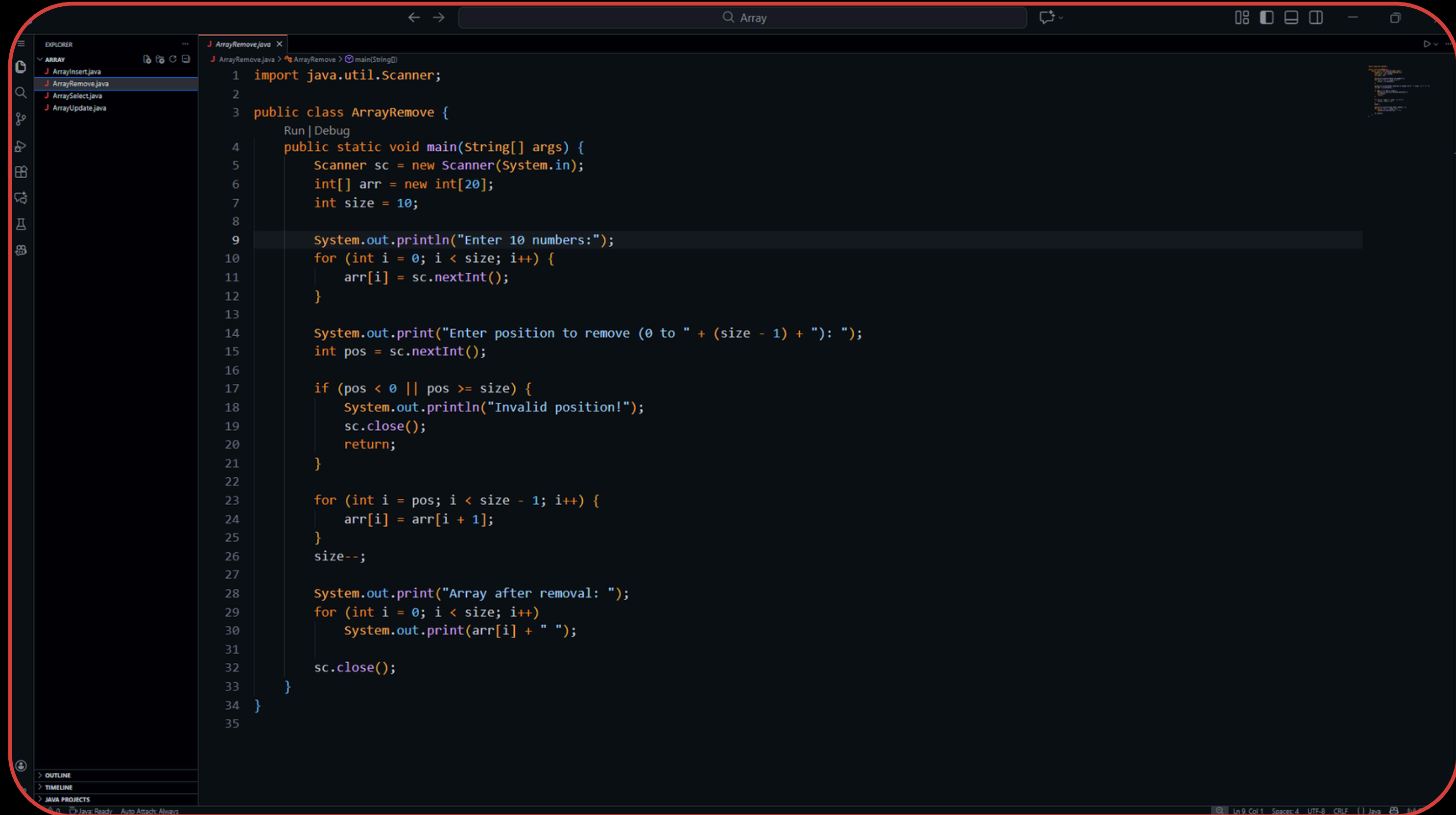


**Remove array[i]  
from array**



#techbywebcoder

# Program



The screenshot shows an IDE with a dark theme. The Explorer panel on the left shows a project named 'ARRAY' with files: 'ArrayInsert.java', 'ArrayRemove.java' (selected), 'ArraySelect.java', and 'ArrayUpdate.java'. The main editor displays the code for 'ArrayRemove.java'. The code imports 'java.util.Scanner', defines a 'public class ArrayRemove', and includes a 'main' method. The 'main' method uses a 'Scanner' to take input, creates an array of size 20, prompts the user to enter 10 numbers, then prompts for a position to remove. It checks for invalid positions and shifts elements to the right if the position is valid, finally printing the array after removal.

```
1 import java.util.Scanner;
2
3 public class ArrayRemove {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[20];
7         int size = 10;
8
9         System.out.println("Enter 10 numbers:");
10        for (int i = 0; i < size; i++) {
11            arr[i] = sc.nextInt();
12        }
13
14        System.out.print("Enter position to remove (0 to " + (size - 1) + "): ");
15        int pos = sc.nextInt();
16
17        if (pos < 0 || pos >= size) {
18            System.out.println("Invalid position!");
19            sc.close();
20            return;
21        }
22
23        for (int i = pos; i < size - 1; i++) {
24            arr[i] = arr[i + 1];
25        }
26        size--;
27
28        System.out.print("Array after removal: ");
29        for (int i = 0; i < size; i++)
30            System.out.print(arr[i] + " ");
31
32        sc.close();
33    }
34 }
35
```

At the bottom of the IDE, there are tabs for 'OUTLINE', 'TIMELINE', and 'JAVA PROJECTS'. The status bar at the very bottom shows 'Ln 9 Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

# Output

```
C:\Windows\System32\cmd.e  ×  +  ▾  
D:\system\DSA\1. Array Operation\Array>javac ArrayRemove.java  
D:\system\DSA\1. Array Operation\Array>java ArrayRemove  
Enter The 10 Number:  
10  
20  
30  
40  
50  
60  
70  
80  
90  
100  
Enter The Position To remove  
7  
Array After Removal :10  
20  
30  
40  
50  
60  
70  
90  
100  
D:\system\DSA\1. Array Operation\Array>
```



# 3.

## SELECT

# Operation In Array

#techbywebcoder

# SELECT Operation In Array



## Algorithm

Step 1: Start

Step 2: Input value to search

Step 3: For  $i = 0$  to  $\text{size}-1$

    If  $\text{arr}[i] = \text{value}$

        Print "Value found at index  $i$ " and stop

Step 4: If loop ends, print "Value not found"

Step 5: Stop

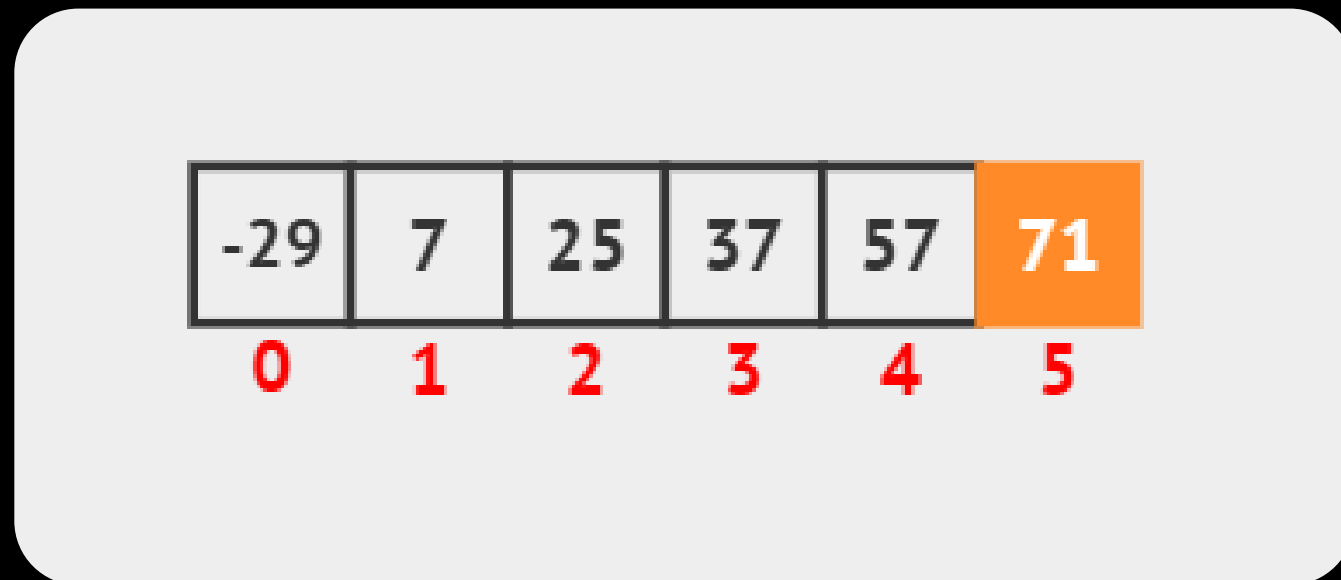
**Select 5 rank item  
in sorted array**

|     |   |    |    |    |    |
|-----|---|----|----|----|----|
| -29 | 7 | 25 | 37 | 57 | 71 |
| 0   | 1 | 2  | 3  | 4  | 5  |

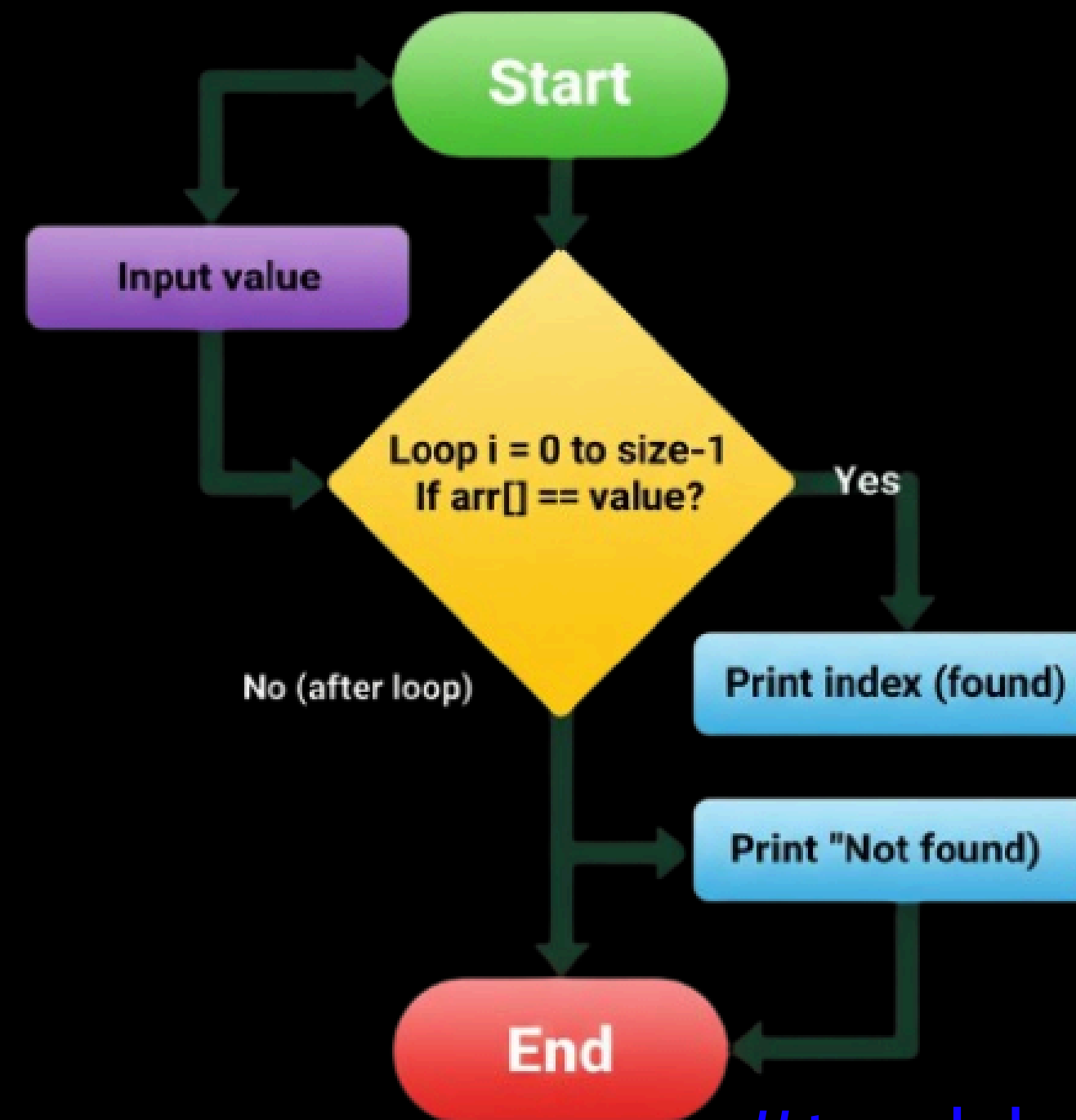
#techbywebcoder

# SELECT Operation In Array

## Flowchart

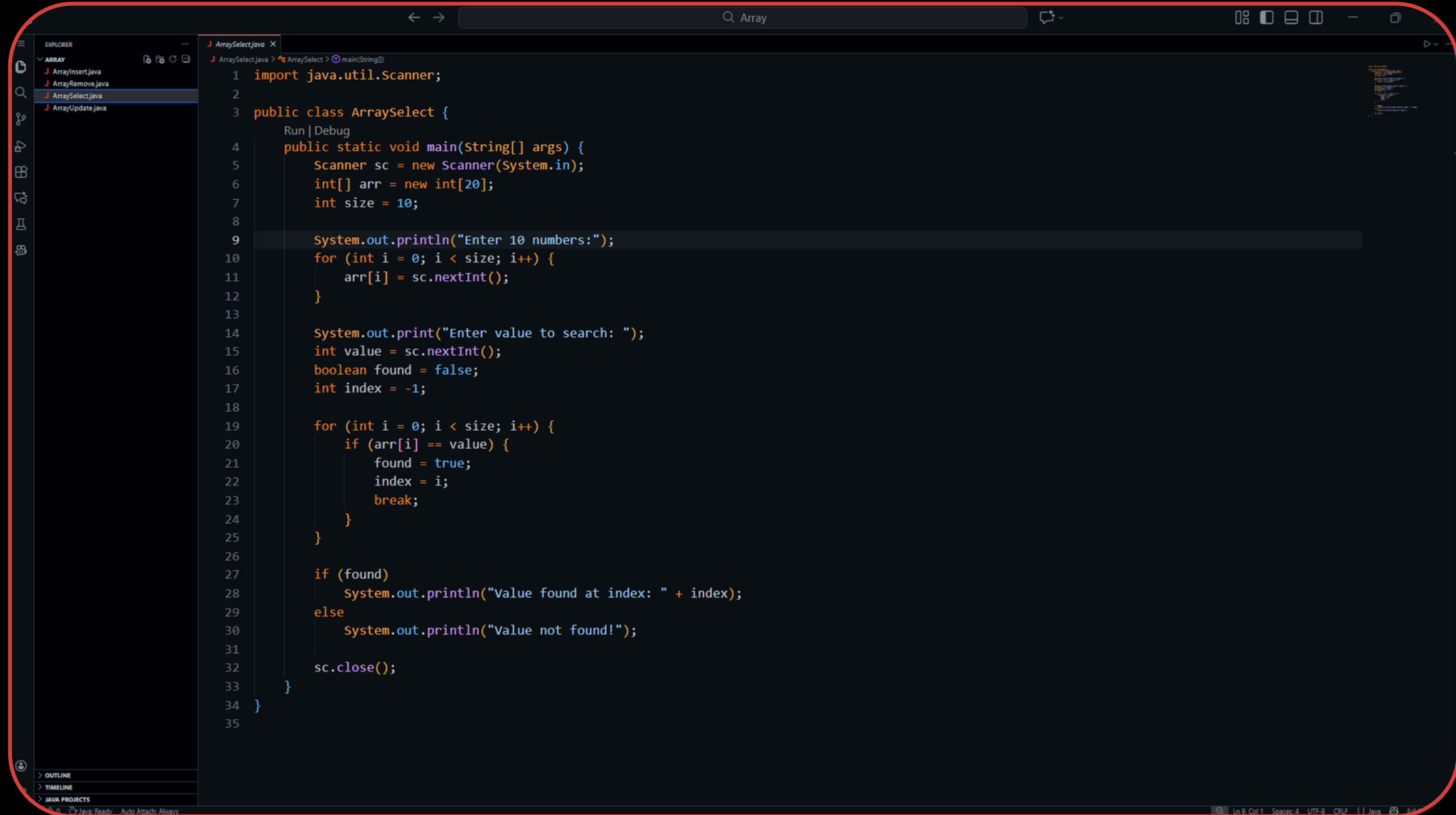


Select 5 rank item  
in sorted array



#techbywebcoder

# Program



The screenshot shows an IDE window with a search bar at the top containing the text "Array". On the left, the Explorer panel shows a project named "ARRAY" with files: "ArrayInsert.java", "ArrayRemove.java", "ArraySelect.java" (selected), and "ArrayUpdate.java". The main editor displays the code for "ArraySelect.java". The code is as follows:

```
1 import java.util.Scanner;
2
3 public class ArraySelect {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[20];
7         int size = 10;
8
9         System.out.println("Enter 10 numbers:");
10        for (int i = 0; i < size; i++) {
11            arr[i] = sc.nextInt();
12        }
13
14        System.out.print("Enter value to search: ");
15        int value = sc.nextInt();
16        boolean found = false;
17        int index = -1;
18
19        for (int i = 0; i < size; i++) {
20            if (arr[i] == value) {
21                found = true;
22                index = i;
23                break;
24            }
25        }
26
27        if (found)
28            System.out.println("Value found at index: " + index);
29        else
30            System.out.println("Value not found!");
31
32        sc.close();
33    }
34 }
35
```

The status bar at the bottom indicates "Ln 9 Col 1", "Spaces: 4", "UTF-8", "CRLF", and "Java".



# Output

```
C:\Windows\System32\cmd.e  ×  +  ▾  
D:\system\DSA\1. Array Operation\Array>javac ArraySelect.java  
D:\system\DSA\1. Array Operation\Array>java ArraySelect  
Enter 10 Numbers:  
10  
20  
30  
40  
50  
60  
70  
80  
  
90  
100  
Enter Value To Search :40  
Value Found At Index:3  
  
D:\system\DSA\1. Array Operation\Array>
```



# 4.

## UPDATE

# Operation In Array

#techbywebcoder

# UPDATE Operation In Array



## Algorithm

Step 1: Start

Step 2: Input position to update

Step 3: If position  $< 0$  OR position  $\geq$  size

Print "Invalid position" and stop

Step 4: Input new value

Step 5: Set `arr[position] = new value`

Step 6: Stop

#techbywebcoder

|    |    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|----|---|
| 13 | 19 | 22 | 13 | 24 | 55 | 62 | 67 |   |
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8 |

Update array  
element at chosen  
index

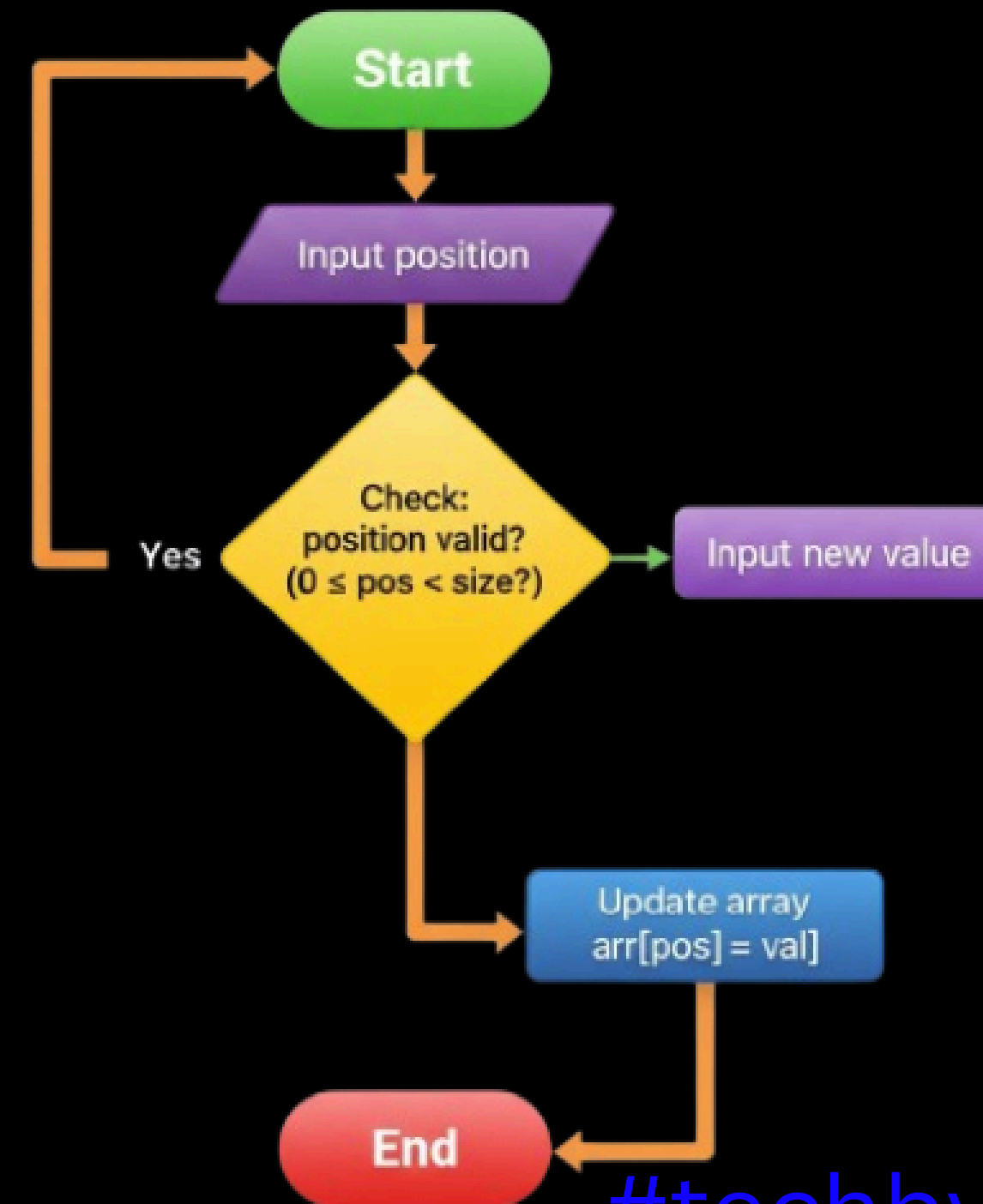
# UPDATE Operation In Array

## Flowchart



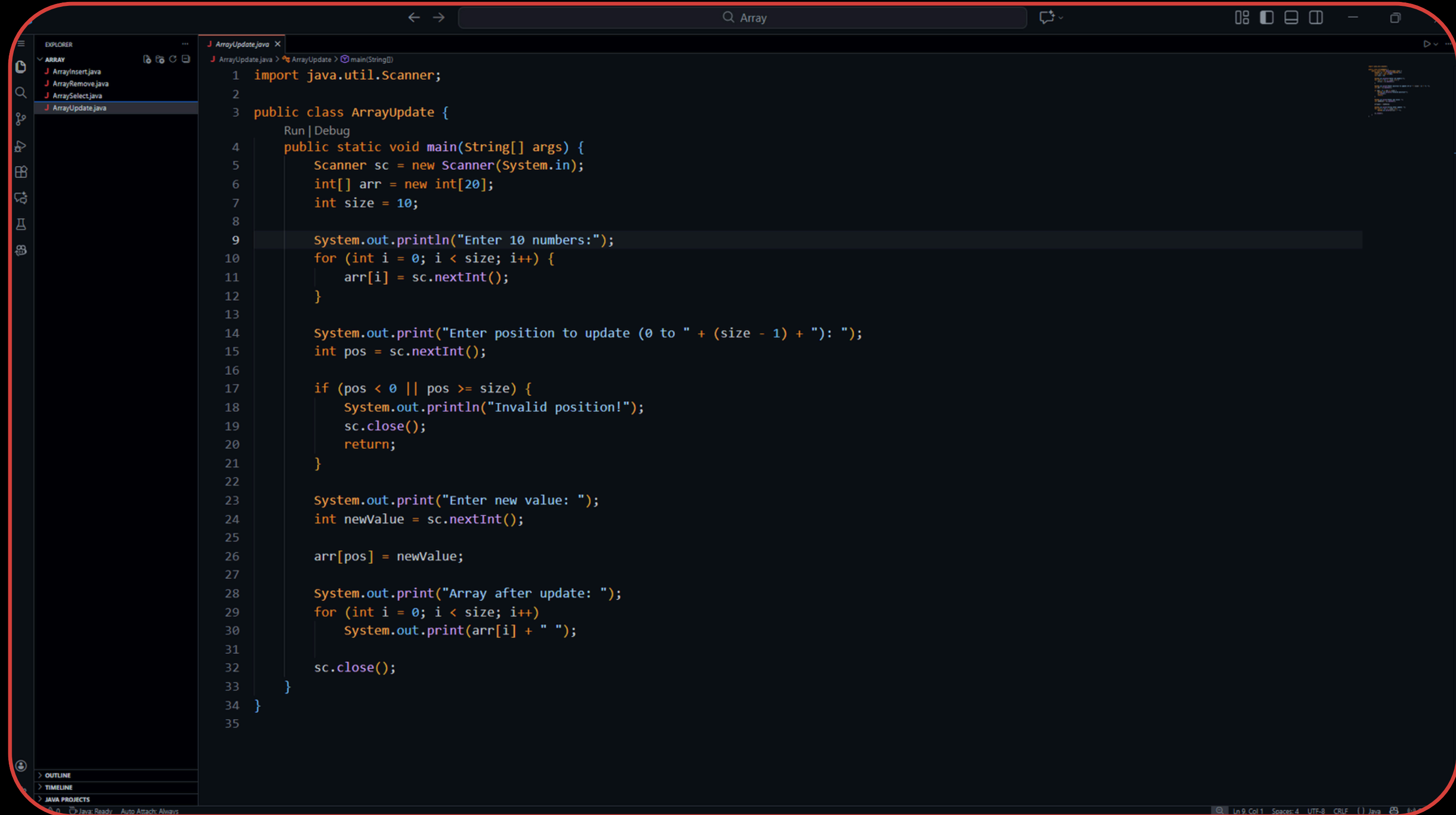
|    |    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|----|---|
| 13 | 19 | 22 | 13 | 24 | 55 | 62 | 67 |   |
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8 |

Update array  
element at chosen  
index



#techbywebcoder

# Program



The screenshot shows an IDE with a dark theme. The Explorer panel on the left shows a project named 'ARRAY' with files: 'ArrayInsert.java', 'ArrayRemove.java', 'ArraySelect.java', and 'ArrayUpdate.java'. The main editor displays the code for 'ArrayUpdate.java'. The code is as follows:


```
1 import java.util.Scanner;
2
3 public class ArrayUpdate {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[20];
7         int size = 10;
8
9         System.out.println("Enter 10 numbers:");
10        for (int i = 0; i < size; i++) {
11            arr[i] = sc.nextInt();
12        }
13
14        System.out.print("Enter position to update (0 to " + (size - 1) + "): ");
15        int pos = sc.nextInt();
16
17        if (pos < 0 || pos >= size) {
18            System.out.println("Invalid position!");
19            sc.close();
20            return;
21        }
22
23        System.out.print("Enter new value: ");
24        int newValue = sc.nextInt();
25
26        arr[pos] = newValue;
27
28        System.out.print("Array after update: ");
29        for (int i = 0; i < size; i++)
30            System.out.print(arr[i] + " ");
31
32        sc.close();
33    }
34 }
35
```

The status bar at the bottom indicates 'Ln 9 Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

# Output

```
C:\Windows\System32\cmd.e  ×  +  ▾  
D:\system\DSA\1. Array Operation\Array>javac ArrayUpdate.java  
D:\system\DSA\1. Array Operation\Array>java ArrayUpdate  
Enter 10 Numbers:  
10  
20  
30  
40  
50  
60  
70  
80  
90  
100  
Enter Position To Update (0 to 9): 6  
Enter New Value: 500  
Array After Update:  
10 20 30 40 50 60 500 80 90 100  
D:\system\DSA\1. Array Operation\Array>
```


Thank you!



**TECH BY  
WEBCODER**

REFACTORING IS LIKE DENTAL HYGIENE FOR  
SOFTWARE ENGINEERS

5  
3  
JS



**Tech By WebCoder**

@TechByWebCoder · 1.85k subscribers · 224 videos

Hello Guys..... ..more

[instagram.com/tech\\_bywebcoder?igsh=YXByM3kwbmZqOGkx](https://www.instagram.com/tech_bywebcoder?igsh=YXByM3kwbmZqOGkx) and 5 more links

🔔 Subscribed ▾

Subscribe For More !