

# Final arrays in Java

Difficulty Level : Easy Last Updated : 31 Oct, 2020

Predict the output of following Java program.

```
class Test
{
    public static void main(String args[])
    {
        final int arr[] = {1, 2, 3, 4, 5}; // Note: arr is final
        for (int i = 0; i < arr.length; i++)
        {
            arr[i] = arr[i]*10;
            System.out.println(arr[i]);
        }
    }
}
```

Output:

```
10
20
30
40
50
```

The array *arr* is declared as final, but the elements of array are changed without any problem. Arrays are objects and object variables are always references in Java. So, when we declare an object variable as final, it means that the variable cannot be changed to refer to anything else. For example, the following program 1 compiles without any error and program 2 fails in compilation.

```
// Program 1
class Test
{
    int p = 20;
    public static void main(String args[])
    {
```

```
{
    final Test t = new Test();
    t.p = 30;
    System.out.println(t.p);
}
```

Output: 30

```
// Program 2
class Test
{
    int p = 20;
    public static void main(String args[])
    {
        final Test t1 = new Test();
        Test t2 = new Test();
        t1 = t2;
        System.out.println(t1.p);
    }
}
```

Output: Compiler Error: cannot assign a value to final variable t1

*So a final array means that the array variable which is actually a reference to an object, cannot be changed to refer to anything else, but the members of array can be modified.*

As an exercise, predict the output of following program

```
class Test
{
    public static void main(String args[])
    {
        final int arr1[] = {1, 2, 3, 4, 5};
        int arr2[] = {10, 20, 30, 40, 50};
        arr2 = arr1;
        arr1 = arr2;
        for (int i = 0; i < arr2.length; i++)
            System.out.println(arr2[i]);
    }
}
```



Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.