

# Tracing Java Programs

**Directions**

1. Take a look at program Test.java below.
2. Read the Java Notes below, and refer to the Test.java program.
3. Trace the program and determine its output, using the trace table provided.

Java Notes

```
int[] array1 = new int[] {3, 67, 4, 12};
```

This is the declaration for an array named **array1**. Notice that the size of the array is not specified with the **new int[]** part of the declaration, but **array1** will be allocated with enough space to contain four integers, and **array1** will be initialized with values 3, 67, 4, and 12. Keep in mind that **array1** will be indexed with **array1[0]** to **array1[3]**.

```
for (int i = 0; i < 4; i++) {
```

One different item to notice about this for loop, is that the loop variable **i** is being declared in the **for** loop. (Noah likes this method ☺) One advantage of declaring loop variables this way is that the "scope" of the variable is only active for the loop – if you try to use the variable after the **}** that ends the loop you will get an undeclared-identifier type error. The variable **i** will be local to the loop.

```
public class Test {

    public static void main(String[] args) {
        int[] array1 = new int[] {3, 67, 4, 12};

        for (int i = 0; i < 4; i++) {
            if (array1[i] > 10) {
                System.out.print(array1[i] + " ");
            }
        }
    } // end main
} // end class Test
```

**Trace Table**

i	array1[i]	array1[i] > 10 ?	Output	Comments/Notes

Note: you can complete the column **array1[i] > 10 ?** with True or False

# Tracing Java Programs

Solution:

i	array1[i]	array1[i] > 10 ?	Output	Comments/Notes
0	3	False	blank	nothing added this time through the loop
1	67	True	67	output 67 because it is > 10
2	5	False	blank	nothing added this time through the loop – you could also say the output contains: 67 and make the output column "cumulative"
3	12	True	12	output because 12 > 10 a cumulative output would be 67 12 because the System.out.print statement does not specify anywhere to use a \n

```

1 public class Test {
2
3
4     public static void main(String[] args) {
5         int[] array1 = new int[] {3, 67, 4, 12};
6
7         for (int i = 0; i < 4; i++) {
8             if (array1[i] > 10) {
9                 System.out.print(array1[i] + " ");
10            }
11        }
12    } // end main

```

```

workspace/ x +
~/workspace $ javac Test.java
~/workspace $ java Test
67 12 ~/workspace $

```