



# Activity 2

*Dr. Ghada Fathy*

# Part A: Q1: Predict output



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    for (int i = 0; i < 5; i++);  
    {  
        System.out.println("Hello");  
    }  
}
```

# Part A: Q1: Predict output



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    for (int i = 0; i < 5; i++);  
    {  
        System.out.println("Hello");  
    }  
}
```

Output  
Hello

Why ??

There is a **semicolon after the for loop**:  
So the loop runs 5 times doing nothing.

## Q2: Predict output



```
for (int i = 0; i < 6; i++) {  
    if (i % 2 == 0)  
        continue;  
  
    System.out.print(i + " ");  
}
```

## Q2: Predict output



```
for (int i = 0; i < 6; i++) {  
    if (i % 2 == 0)  
        continue;  
  
    System.out.print(i + " ");  
}
```

Output

1 3 5

When i is even → continue skips print.

Only odd numbers are printed.

## Q3. Predict output



```
for (int i = 1; i <= 3; i++) {  
    for (int j = 1; j <= 3; j++) {  
        if (j == 2)  
            break;  
        System.out.print(i + "" + j + " ");  
    }  
}
```

## Q3. Predict output



```
for (int i = 1; i <= 3; i++) {  
    for (int j = 1; j <= 3; j++) {  
        if (j == 2)  
            break;  
        System.out.print(i + "" + j + " ");  
    }  
}
```

Output

11

21

31

## Q4. Predict output

```
int[] arr = {10, 20, 30};  
System.out.println(arr[3]);
```



## Q4. Predict output



```
int[] arr = {10, 20, 30};  
System.out.println(arr[3]);
```

### Output

Runtime Error:

ArrayIndexOutOfBoundsException

## Q5. Predict output

```
int[] arr = {5, 10, 15};  
  
for (int i = arr.length - 1; i >= 0; i--) {  
    System.out.print(arr[i] + " ");  
}
```



## Q5. Predict output

```
int[] arr = {5, 10, 15};  
  
for (int i = arr.length - 1; i >= 0; i--) {  
    System.out.print(arr[i] + " ");  
}
```

Output

15 10 5



## Q6. Predict output

```
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        System.out.print(i + j + " ");  
        i++;  
    }  
}
```

# Q6. Predict output

```
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        System.out.print(i + j + " ");  
        i++;  
    }  
}
```

Output  
0 2 4

## Q7. Predict output

```
int[] arr = {5, 10, 15};

for (int i = 0; i < arr.length; i++) {
    arr[i] = arr[i] + 5;
}

for (int x : arr)
    System.out.print(x + " ");
```

## Q7. Predict output

```
int[] arr = {5, 10, 15};

for (int i = 0; i < arr.length; i++) {
    arr[i] = arr[i] + 5;
}

for (int x : arr)
    System.out.print(x + " ");
```

10 15 20

## Q8. Predict output

```
int[] arr = {1, 2, 3};

for (int x : arr) {
    x = x * 2;
}

for (int x : arr)
    System.out.print(x + " ");
```

# Q8. Predict output

Enhanced for-loop works on copy of element, not original array.  
So original values do NOT change.

```
int[] arr = {1, 2, 3};

for (int x : arr) {
    x = x * 2;
}

for (int x : arr)
    System.out.print(x + " ");
```

1 2 3

## Q9. Correct the following Code

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] arr = {10, 20, 30, 40};  
  
        for (int i = 0; i <= arr.length; i++) {  
            System.out.println(arr[i]);  
        }  
    }  
}
```

# Q9. Correct the following Code

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] arr = {10, 20, 30, 40};  
  
        for (int i = 0; i <= arr.length; i++) {  
            System.out.println(arr[i]);  
        }  
  
    }  
}
```

## Run Time Error

arr.length = 4

Valid indexes: 0 → 3

Using <= arr.length tries to access index 4

That causes: ArrayIndexOutOfBoundsException

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] arr = {10, 20, 30, 40};  
  
        for (int i = 0; i < arr.length; i++) {  
            System.out.println(arr[i]);  
        }  
  
    }  
}
```

## Q10. Correct the following Code

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] numbers;  
        numbers = {1, 2, 3, 4};  
  
        System.out.println(numbers[2]);  
  
    }  
}
```

# Q10. Correct the following Code

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] numbers;  
        numbers = {1, 2, 3, 4};  
  
        System.out.println(numbers[2]);  
  
    }  
}
```

Compilation error.

You can use {} directly only during declaration:

```
numbers = new int[]{1, 2, 3, 4};
```

```
public class Test {  
    public static void main(String[] args) {  
  
        int[] numbers;  
        numbers = new int[]{1, 2, 3, 4};  
  
        System.out.println(numbers[2]);  
  
    }  
}
```

A close-up photograph of a computer keyboard. The central focus is a large, rectangular key with a vibrant blue background and the words "Thank You" printed in a clean, white, sans-serif font. The key is slightly raised and has rounded corners. Surrounding this key are several other standard grey keys, which are slightly out of focus, creating a sense of depth. The lighting is soft and even, highlighting the texture of the plastic keys and the smooth surface of the blue key.

*Thank You*