



Java: Loops and Array

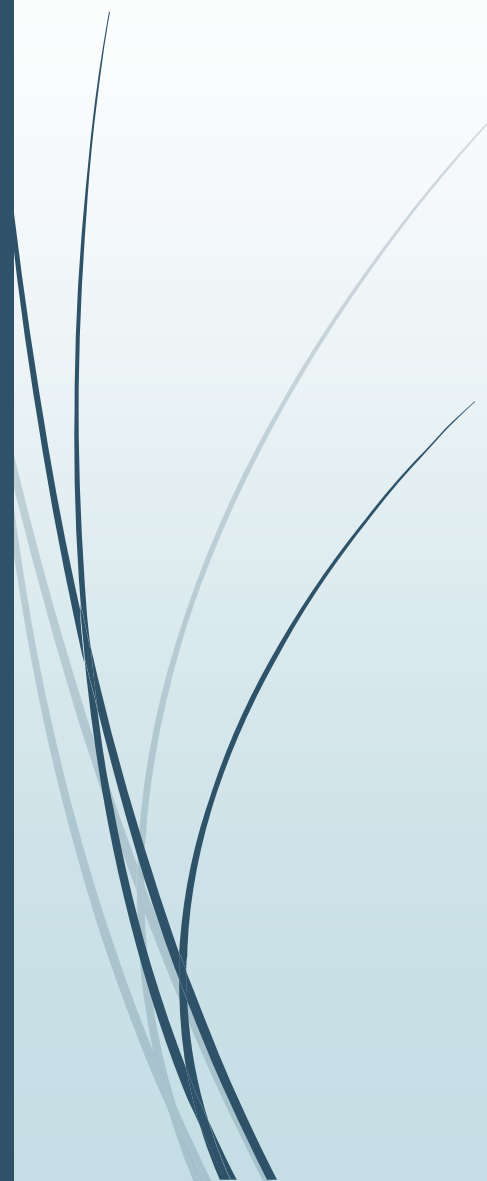
B.Sc. 2nd Semester

Department of Computer Science and Applications (Paper Code:CC3)

Prof. Paulami Basu Ray



Contents

- ▶ Loops
 - ▶ Arrays
- 

A dark blue arrow points to the right at the top left. Several thin, curved lines in shades of blue and grey sweep across the left side of the slide.

Loops/ Iterations

- I. Looping in programming languages is a feature which facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true.
- II. Different types of loops available in Java:
 - while loop
 - do-while loop
 - for loop
 - Enhanced for loop (for each style loop)

A dark blue arrow points to the right at the top left. Several thin, curved lines in shades of blue and grey sweep across the left side of the slide.

While loop

► Syntax:

Initialization;

while (testing condition)

{

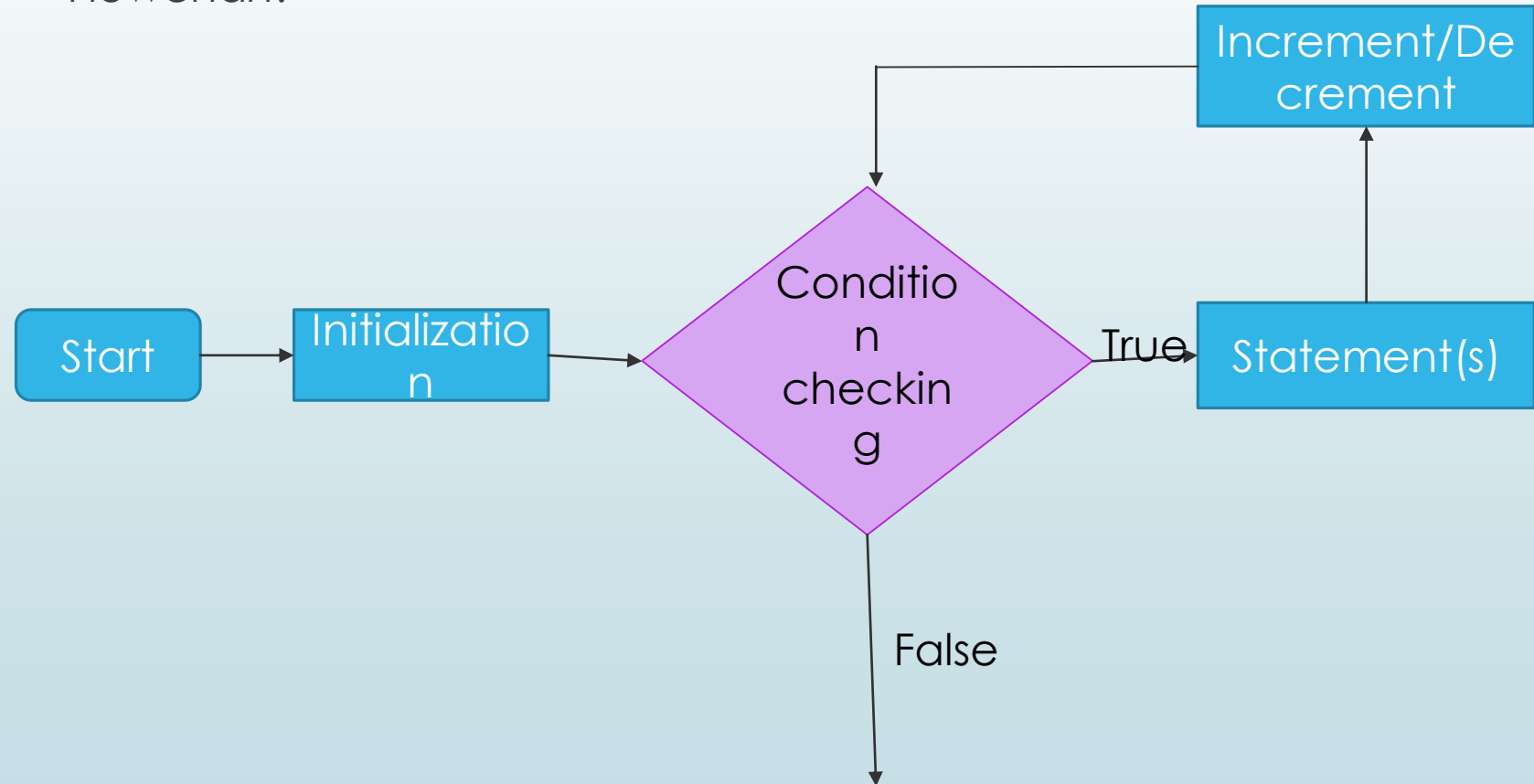
 statement(s)

 increment/decrement;

}

While loop

► Flowchart:



A dark blue arrow points to the right from the left edge of the slide. Several thin, curved lines in shades of blue and grey originate from the left side and sweep across the slide towards the right.

do-while loop

► Syntax:

```
Initialization;
```

```
do
```

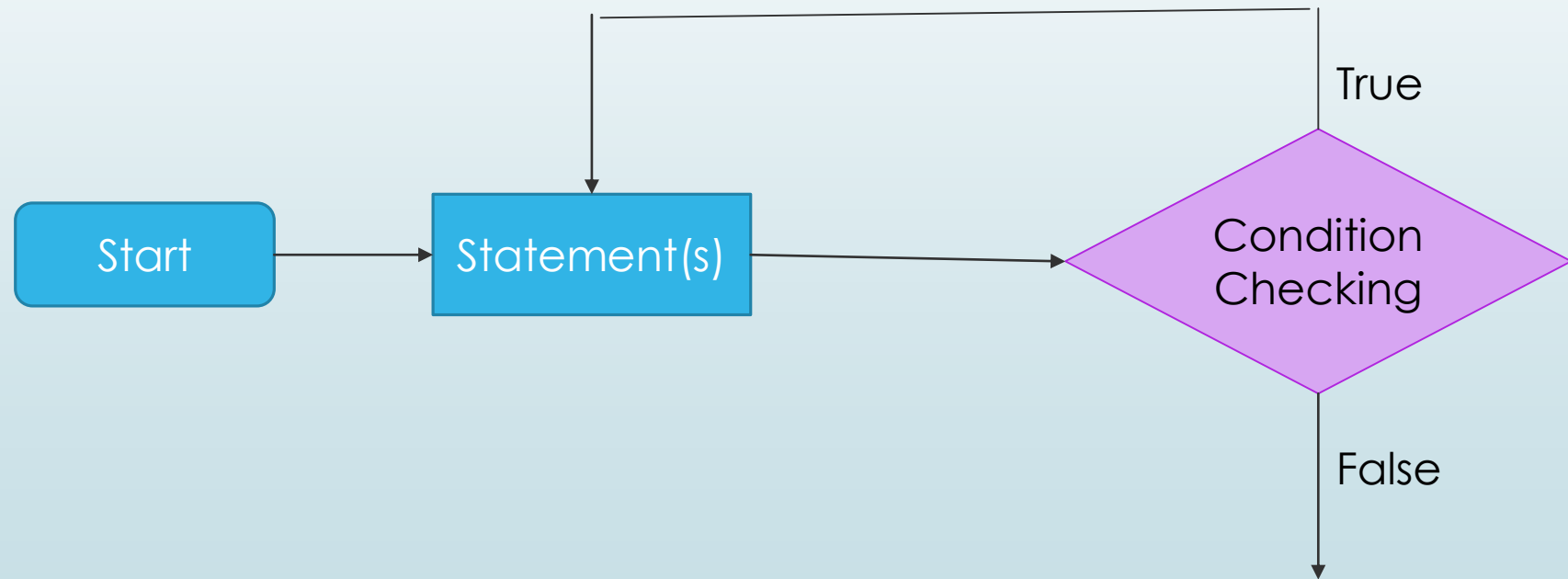
```
{  statement(s)
```

```
    increment/decrement;
```

```
} while ( testing condition);
```

do-while loop

► Flowchart

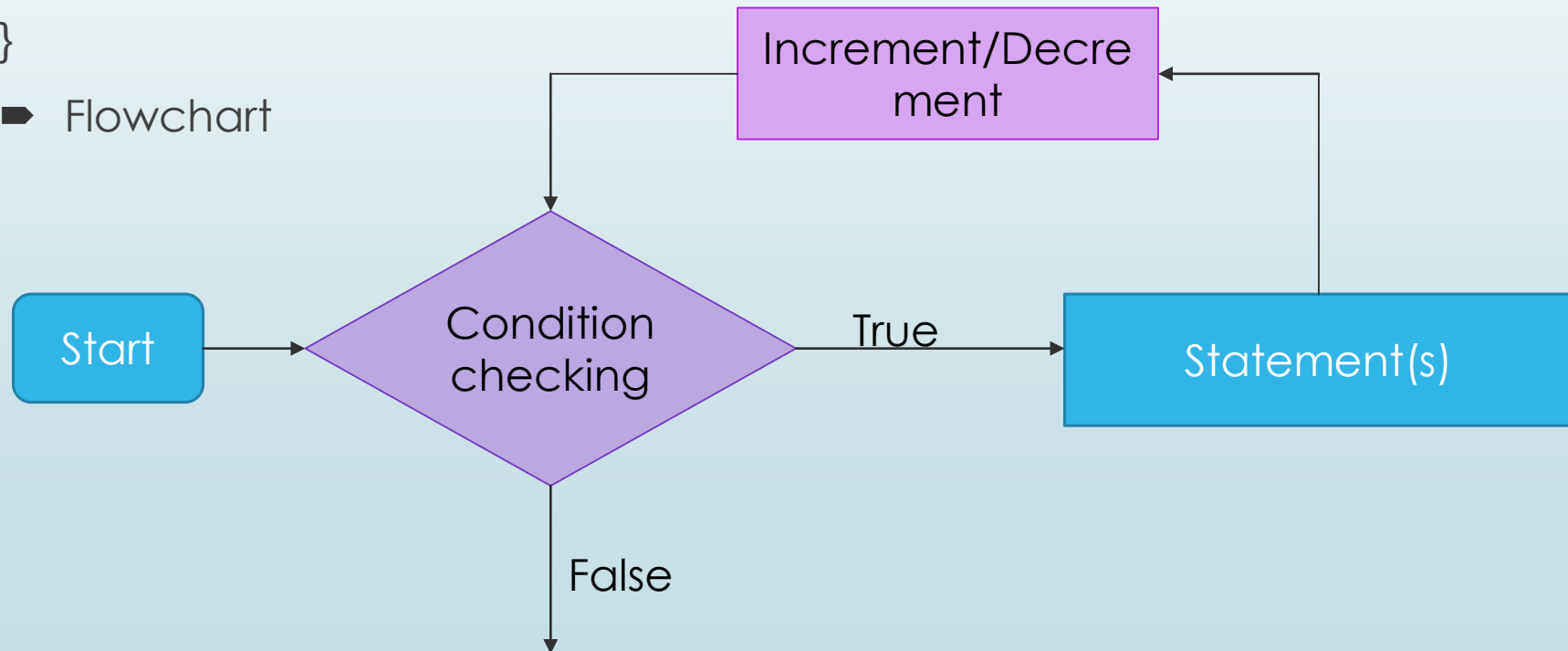


For loop

► Syntax:

```
for ( initialization; testing condition; increment/decrement )  
{  
    statement(s)  
}
```

► Flowchart





Enhanced for loop

► Syntax:

```
for (datatype element: collection)
```

```
{
```

```
    statement(s)
```

```
}
```

break - continue

```
public class BreakContinueExample {  
    public static void main(String args[]) {  
        //calculate sum of all numbers till 5 appears  
        int sum = 0;  
  
        for(int i=0; i< 10; i++){  
            if(i == 5){  
                System.out.println("calling break statement to break for loop");  
                break;  
            }  
            if(i %2 != 0){  
                sum = sum + i;  
                continue;  
            }  
            System.out.println("Last line of loop, not executing for odd numbers due  
                to continue statement i: " + i);  
        }  
        System.out.println("Outside of for loop, sum: " + sum);  
    }  
}
```



break-continue

Output:

Last line of loop, not executing for odd numbers due to **continue** statement i: 0

calling **continue** statement to start **new** iteration

Last line of loop, not executing for odd numbers due to **continue** statement i: 2

calling **continue** statement to start **new** iteration

Last line of loop, not executing for odd numbers due to **continue** statement i: 4

calling **break** statement to **break** for loop

Outside of for loop, sum: 4



Arrays

- ▶ In Java all arrays are dynamically allocated.
- ▶ Since arrays are objects in Java, we can find their length using member length. This is different from C/C++ where we find length using sizeof.
- ▶ A Java array variable can also be declared like other variables with [] after the data type.
- ▶ The variables in the array are ordered and each have an index beginning from 0.

Arrays

- **One-dimensional array:**

- Syntax:

```
data-type array-name[];
```

OR

```
data-type[] array-name;
```

- **Multi-dimensional array:**

```
int[][] intArray = new int[10][20];
```

```
Int[][][] intArray = new int[10][20][30];
```



Length of 2 Dimensional Array

```
public class TwoDArrayDemo {  
  
    public static void main(String args[]) {  
  
        int[][] test; test = new int[5][10];  
        int row = test.length;  
        int col = test[0].length;  
        System.out.println(row);  
        System.out.println(col);  
  
    }  
}
```