**Before class 3**

**1.**

**Static field**

When a number of objects are created from the same class blueprint, they each have their own distinct copies of instance variables. In the case of the Bicycle class, the instance variables are cadence, gear, and speed. Each Bicycle object has its own values for these variables, stored in different memory locations. Sometimes, you want to have variables that are common to all objects. This is accomplished with the static modifier. Fields that have the static modifier in their declaration are called static fields or class variables. They are associated with the class, rather than with any object. Every instance of the class shares a class variable, which is in one fixed location in memory. Any object can change the value of a class variable, but class variables can also be manipulated without creating an instance of the class.

**Instance field**

->The fields which are defined within class without static modifiers.

->Every instant field itself is having its own corresponding values.

->For the instance fields memory is allocation within the object.

->Instance fields are always referenced by using an object.

->Another name of instance fields are non-static fields.

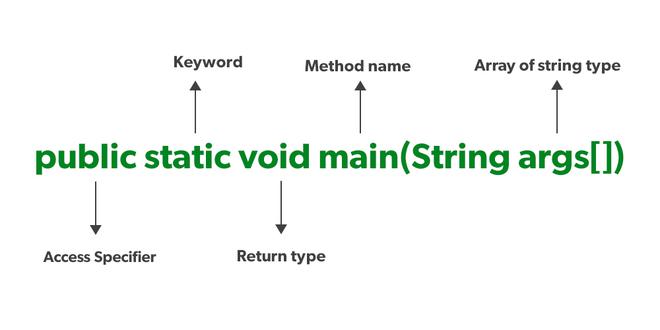
**Static method**

The static keyword is used to construct methods that will exist regardless of whether or not any instances of the class are generated. Any method that uses the static keyword is referred to as a static method.

* Features of static method:
  + A static method in Java is a method that is part of a class rather than an instance of that class.
  + Every instance of a class has access to the method.
  + Static methods have access to class variables (static variables) without using the class’s object (instance).
  + Only static data may be accessed by a static method. It is unable to access data that is not static (instance variables).
  + In both static and non-static methods, static methods can be accessed directly.

**2.**

In Java programs, the point from where the program starts its execution or simply the entry point of Java programs is the main() method, in the end of the class.



**3.**

In class-based, object-oriented programming, a constructor is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.

**4.**

* In Java, all arrays are dynamically allocated.
* Arrays are stored in contagious memory [consecutive memory locations].
* Since arrays are objects in Java, we can find their length using the object property length.
* 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 has an index beginning from 0.
* Java array can also be used as a static field, a local variable, or a method parameter.
* The size of an array must be specified by int or short value and not long.
* The direct superclass of an array type is Object.
* Every array type implements the interfaces Cloneable and java.ioSerializable.
* This storage of arrays helps us randomly accessing the elements of an array
* The size of the array cannot be altered (once initialized).
* However, an array reference can be made to point to another array.

An array can contain primitives (int, char, etc.) and object (or non-primitive) references of a class depending on the definition of the array. In the case of primitive data types, the actual values are stored in contiguous memory locations. In the case of class objects, the actual objects are stored in a heap segment.

**One-Dimensional Arrays:**

The general form of a one-dimensional array declaration is

type var-name[];

OR

type[] var-name;

int intArray[];

or int[] intArray;

byte byteArray[];

short shortsArray[];

boolean booleanArray[];

long longArray[];

float floatArray[];

double doubleArray[];

char charArray[];

nstantiating an Array in Java

When an array is declared, only a reference of an array is created. To create or give memory to the array, you create an array like this: The general form of new as it applies to one-dimensional arrays appears as follows:

var-name = new type [size];

Here, type specifies the type of data being allocated, size determines the number of elements in the array, and var-name is the name of the array variable that is linked to the array. To use new to allocate an array, you must specify the type and number of elements to allocate.

Example:

int intArray[]; //declaring array

intArray = new int[20]; // allocating memory to array

or

int[] intArray = new int[20]; // combining both statements in one

Array Literal

In a situation where the size of the array and variables of the array are already known, array literals can be used.

int[] intArray = new int[]{ 1,2,3,4,5,6,7,8,9,10 };

// Declaring array literal

The length of this array determines the length of the created array.

There is no need to write the new int[] part in the latest versions of Java.

Accessing Java Array Elements using for Loop

Each element in the array is accessed via its index. The index begins with 0 and ends at (total array size)-1. All the elements of array can be accessed using Java for Loop.

// accessing the elements of the specified array

for (int i = 0; i < arr.length; i++)

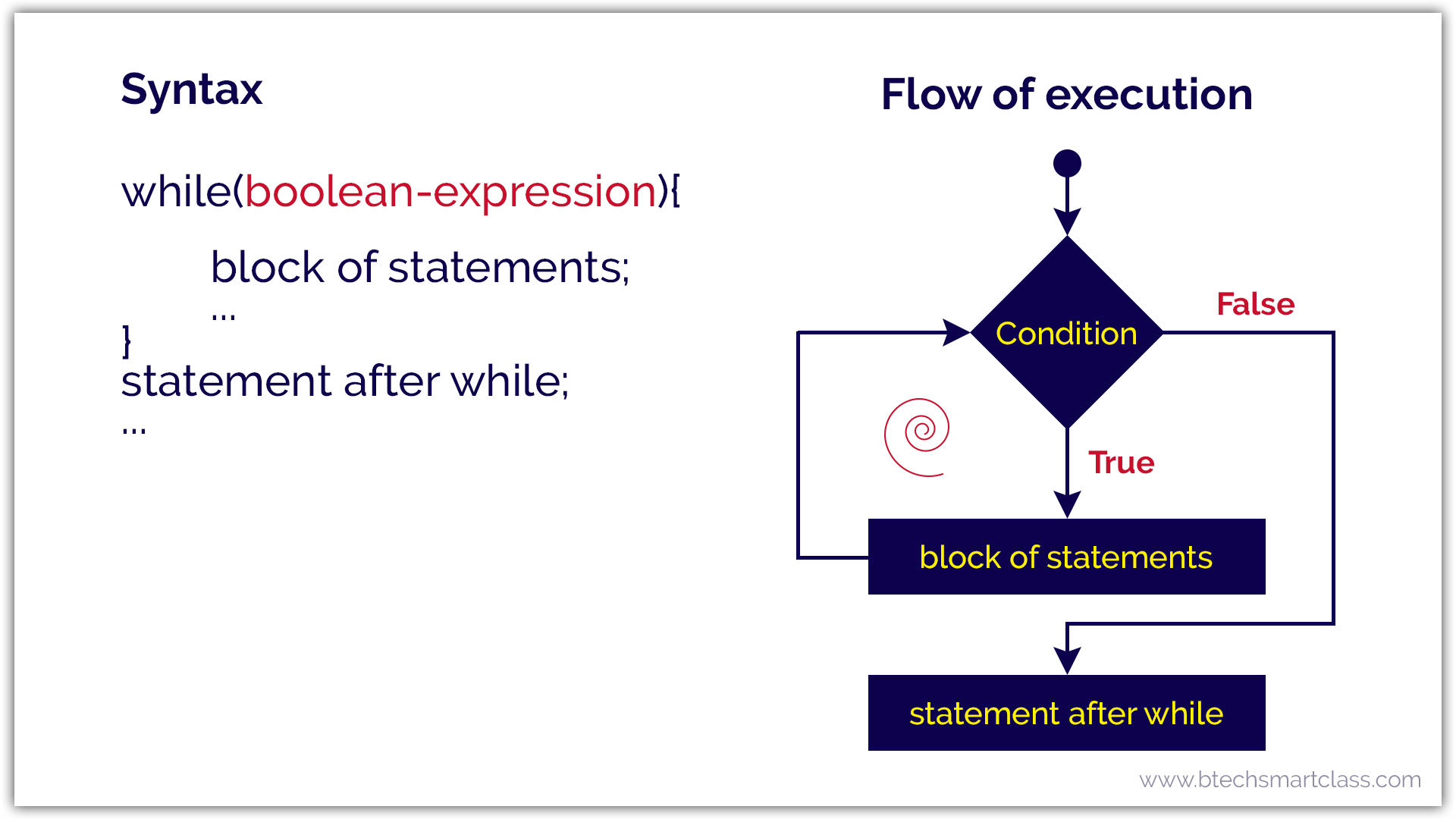
System.out.println("Element at index " + i +

" : "+ arr[i]);

5.

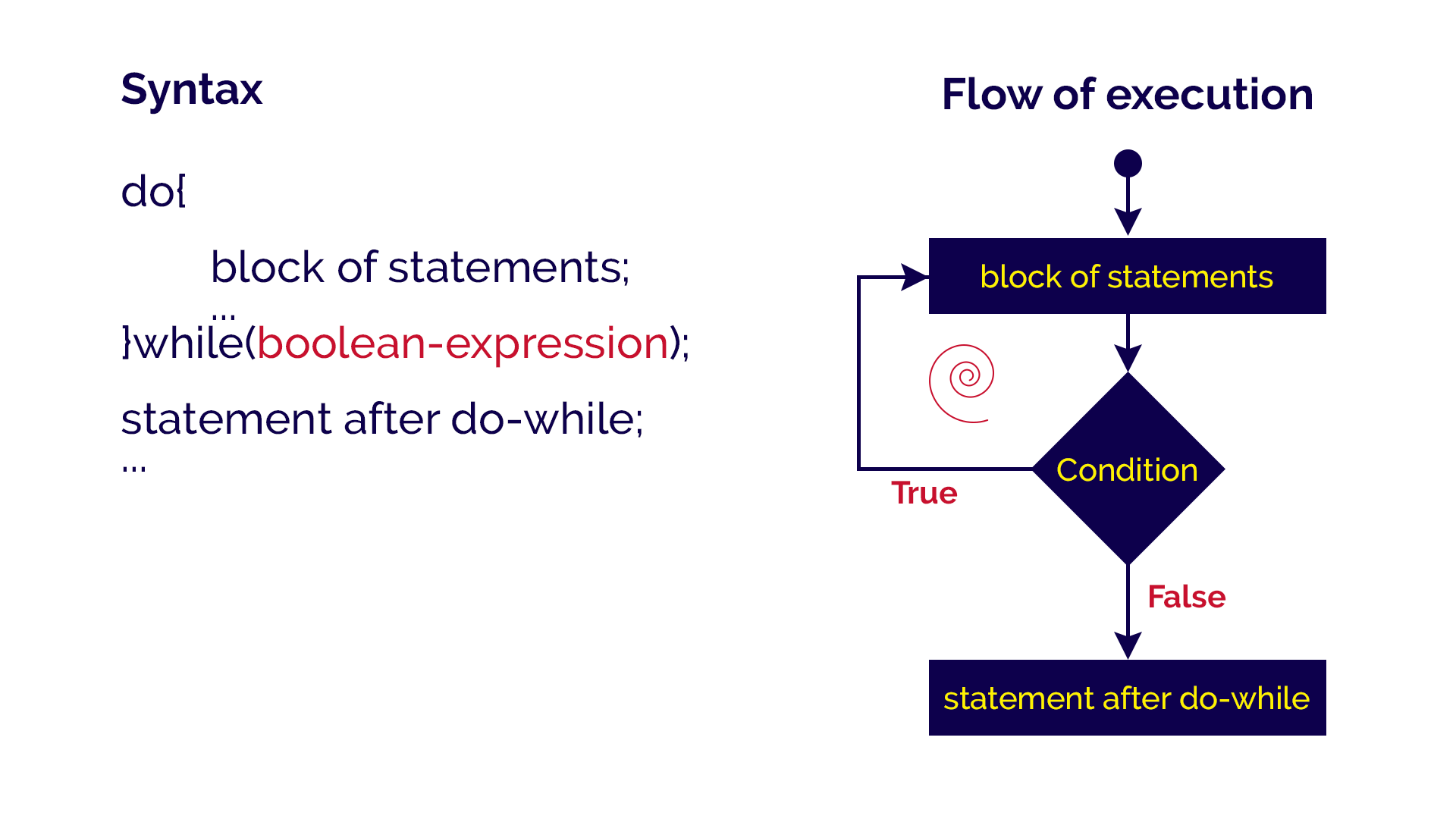
**while statement in java**

The while statement is used to execute a single statement or block of statements repeatedly as long as the given condition is TRUE. The while statement is also known as Entry control looping statement. The syntax and execution flow of while statement is as follows.



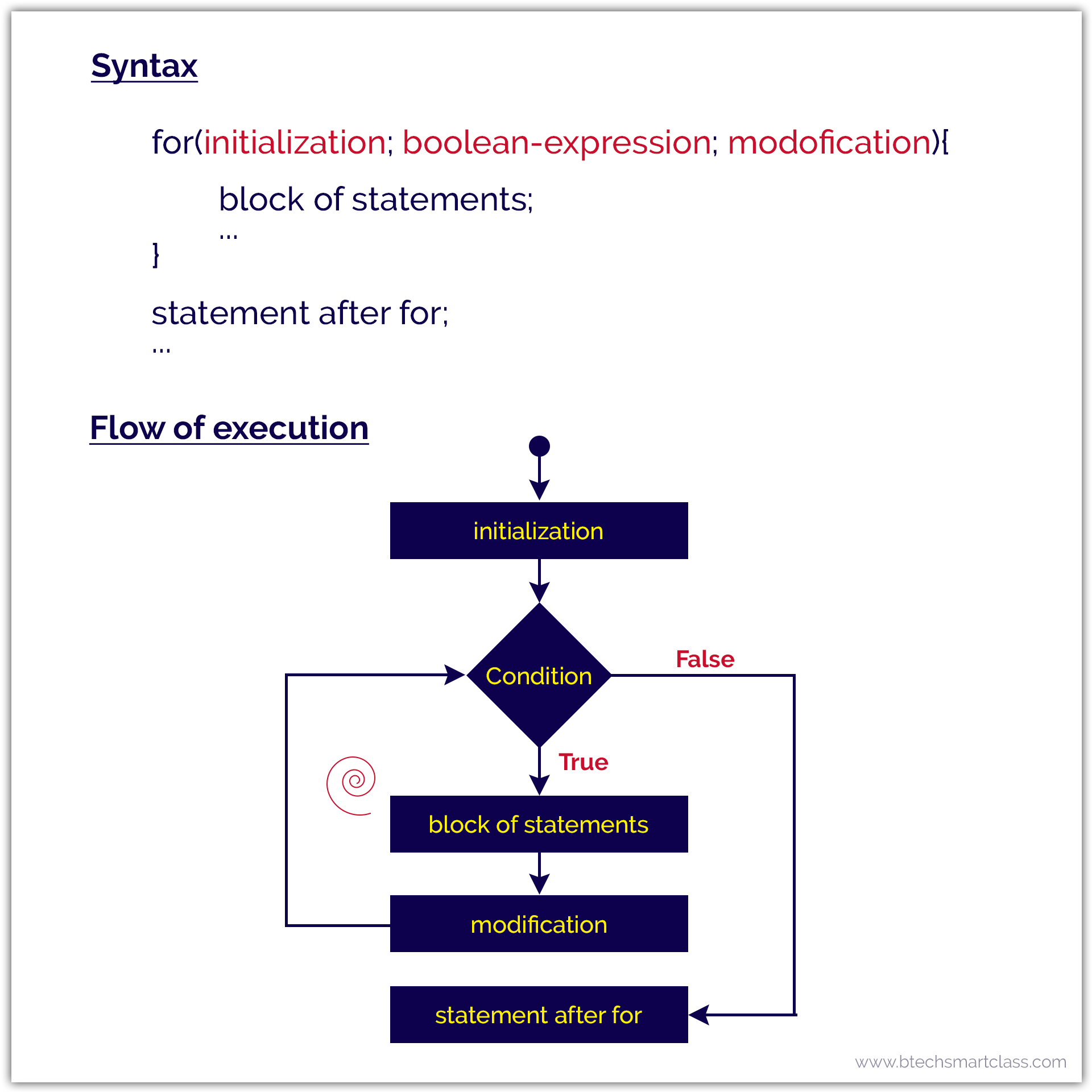
**do-while statement in java**

The do-while statement is used to execute a single statement or block of statements repeatedly as long as given the condition is TRUE. The do-while statement is also known as the **Exit control looping statement**. The do-while statement has the following syntax.



**for statement in java**

The for statement is used to execute a single statement or a block of statements repeatedly as long as the given condition is TRUE. The for statement has the following syntax and execution flow diagram.



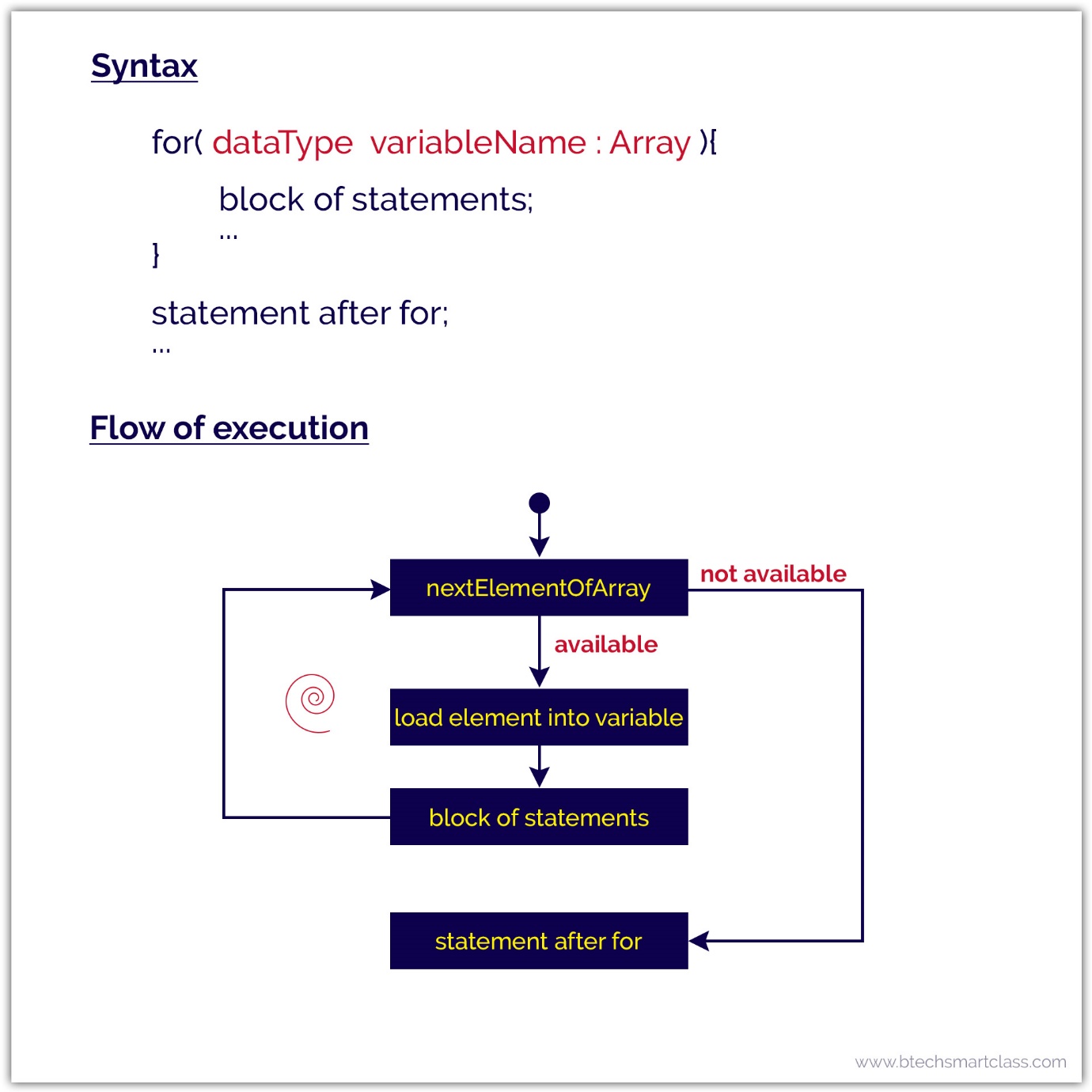
In for-statement, the execution begins with the initialization statement. After the initialization statement, it executes Condition. If the condition is evaluated to true, then the block of statements executed otherwise it terminates the for-statement. After the block of statements execution, the modification statement gets executed, followed by condition again.

**for-each statement in java**

The Java for-each statement was introduced since Java 5.0 version. It provides an approach to traverse through an array or collection in Java. The for-each statement also known as **enhanced for** statement. The for-each statement executes the block of statements for each element of the given array or collection.

🔔 In for-each statement, we can not skip any element of given array or collection.

The for-each statement has the following syntax and execution flow diagram.



## The if Statement

Use the if statement to specify a block of Java code to be executed if a condition is true.

### **Syntax**

if (condition) {

// block of code to be executed if the condition is true

}

## The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

### **Syntax**

if (condition) {

// block of code to be executed if the condition is true

} else {

// block of code to be executed if the condition is false

}

## The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

### **Syntax**

if (condition1) {

// block of code to be executed if condition1 is true

} else if (condition2) {

// block of code to be executed if the condition1 is false and condition2 is true

} else {

// block of code to be executed if the condition1 is false and condition2 is false

}

6.

Method Overloading in Java

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

If we have to perform only one operation, having same name of the methods increases the readability of the program.

1. Method Overloading: changing no. of arguments
2. 2) Method Overloading: changing data type of arguments