Understanding Java Fundamentals: Getters, Setters, Anonymous Blocks, and Data Member Power

Hello everybody,

**Recap: Data Members and Their Importance**

We already learned about data members (variables) within a class and their role in storing object information. Now, let's explore how they interact with constructors and anonymous blocks.

Now in this article we will understand its associativity powers

class Student {

int id=1;

String name=”binod”;

static String course = "MCA";

// Default constructor

Student() {

System.out.println("In default constructor");

System.out.println("Id"+id);

System.out.println("Name"+name);

id=103;

name=”Gaurav”;

}

// Set method to modify instance variables

public void setData(int id, String name) {

this.id = id;

this.name = name;

}

public class Main {

public static void main(String[] args) {

Student s1 = new Student();

}

}

**Output:**

In default constructor

Id=1, Name=binod, Course=MCA

Id=103, Name=Gaurav, Course=MCA

Now as I was fiddling through this code I understood the power of default assigning of values in data members

Even if I don’t assign values the default constructor values should be assigned but NO

The default values which I assigned are getting allocated First then the default constructor values are

Getting assigned first So this is the power of default assignment of values in JAVA

Anonymous blocks

Now this is a whole new concept introduced by JAVA

**Anonymous blocks** are code snippets that are executed automatically even before the constructors  
Real life example of Anonmymous blocks

**Imagine a classroom full of students.** Before the teacher starts the lesson, they might ask everyone to take out their textbooks and sharpen their pencils. This is like an anonymous block. It's a set of tasks that everyone needs to do before the main activity (the lesson) can begin.

**In Java, anonymous blocks are like these "startup tasks" for a class.**

**Code Example with Anonymous Blocks**

class Demo {

Demo() {

System.out.println("In default constructor");

}

{ // Anonymous block

System.out.println("First Anonymous block");

}

public static void main(String[] args) {

Demo b1 = new Demo();

}

}

**Output:**

**First Anonymous block**

**In default constructor**

As you can see, the anonymous block executes before the default constructor.

Now the next question is that

First default Assignment of data members will be executed of anonymous blocks?

The answer is default assignment WHY?

Lets understand it from this small code snippet

class Demo{

int id=100

Demo(){

System.out.println("Id":+id);

}

{

System.out.print("Id :"+id);

Id=1001;

}

public static void main(String arr[]){

Demo b1 = new Demo();

} }

**Output:**

Id: 100 (default assignment)

Id: 1001 (anonymous block assignment)

As you can see from the above code that first default assignment is done to ID

Then anonymous block assignment is done

**USE CASE**

**Imagine you have a company with five employees. Only three require user input, while the remaining two have pre-defined information. Default data member assignment is useful here because you can provide default values for those two workers within the class, avoiding the requirement for user input**

Getters and Setters (coming soon!)  
  
In subsequent articles, we'll look at getters and setters, which are specific methods for accessing and modifying data member values safely.