19-Anonymous Object

Anonymous Object in Java

Object Creation: The Traditional Way

In Java, the traditional way of creating an object involves two main steps:

1. **Declaration of an Object Reference**: First, you declare a reference variable that will point to the object.
2. **Object Creation**: Then, you use the **new keyword** to create the object in the heap memory and assign it to the reference variable.

This approach involves allocating memory in both the stack (for the reference variable) and the heap (for the actual object).

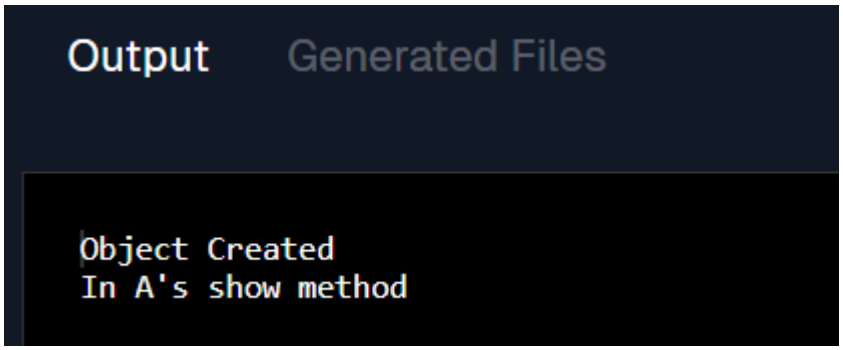
Example:

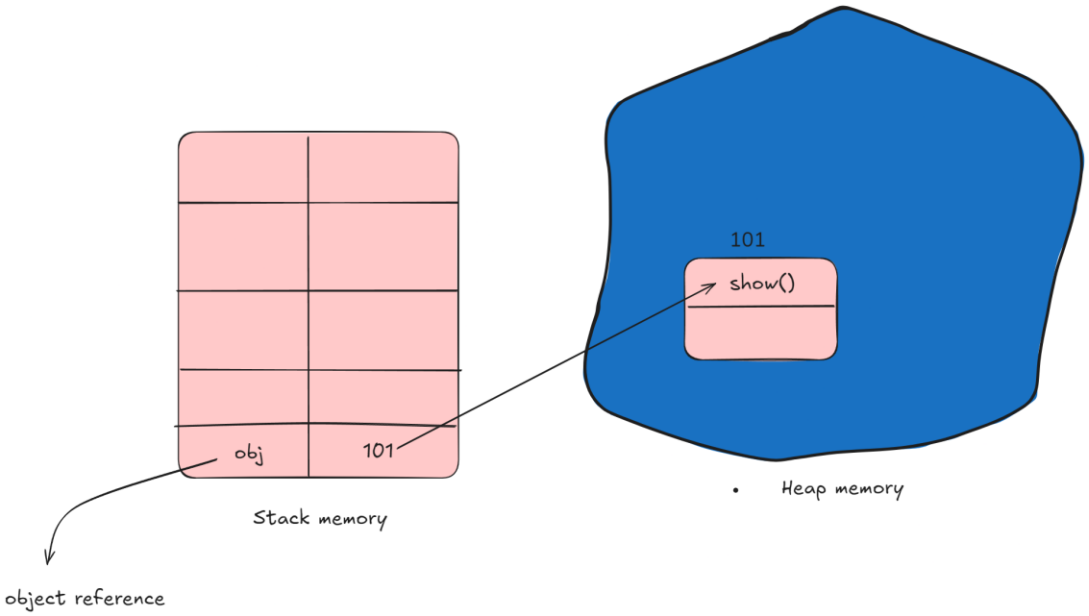
```
class A {
    public A() {
        System.out.println("Object Created");
    }

    public void show() {
        System.out.println("In A's show method");
    }
}

public class Demo {
    public static void main(String[] args) {
        A obj = new A(); // Traditional object creation
        obj.show();
    }
}
```

Output:





Anonymous Object in Java

But what if you want to create an object without allocating space in the stack for the reference variable? Can you directly use an object without assigning it a name or reference?

Yes, you can create such an object, known as an **Anonymous Object**. An anonymous object is created without a reference variable and can still be used to invoke methods.

- In Java, an anonymous object is an object that is created without giving it a name. Anonymous objects are often used to create objects on the fly and pass them as arguments to methods.

Example:

```
class A {
    public A() {
        System.out.println("Object Created");
    }

    public void show() {
        System.out.println("In A's show method");
    }
}

public class Demo {
    public static void main(String[] args) {
        new A().show(); // Anonymous object creation
    }
}
```

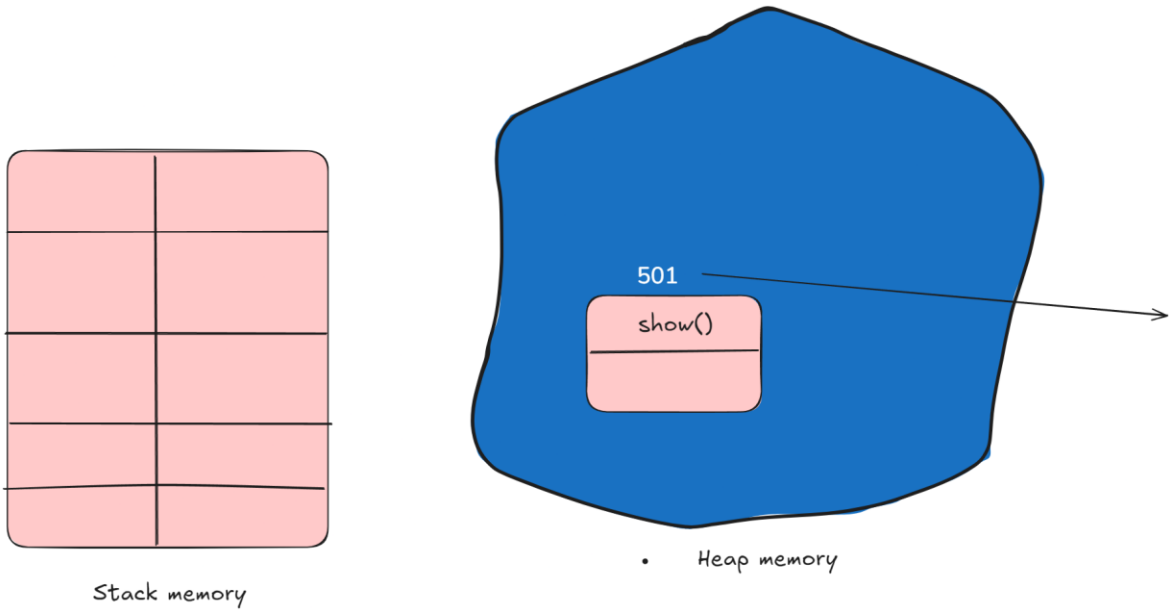
Output:

```
Output      Generated Files

Object Created
In A's show method
```

Explanation of the Example

In the second example, the object is created without assigning it to any reference variable. The new A().show(); statement creates the object of class A, immediately calls its show() method, and then the object becomes eligible for garbage collection once the method execution is complete. This type of object is useful when you need to create an object for one-time use, where retaining the object reference isn't necessary.



Advantages of Using Anonymous Objects

- **Memory Efficiency**: Since no reference is stored in the stack, anonymous objects save memory.
- **Cleaner Code**: Ideal for situations where the object is only used once, resulting in less clutter in the code.

Disadvantages of Using Anonymous Objects

- **No Reusability**: Since the object has no reference, it can't be reused in the program.
- **Immediate Garbage Collection**: The object is eligible for garbage collection right after its usage, which may not be desirable in some cases.