Module 2: Object-Oriented Design: Classes, Inheritance, and Polymorphism

Premanand S

Assistant Professor
School of Electronics Engineering
Vellore Institute of Technology
Chennai Campus

premanand.s@vit.ac.in

February 16, 2026

Class vs Object vs Reference

Key Definitions

- **Class**: Blueprint or template
- **Object**: Real instance created in memory
- **Reference variable**: Stores address of object

Important

Reference variable is NOT the object.

Basic Object Declaration Syntax

Syntax:

```
ClassName reference = new ClassName();
```

Example:

```
Student s1 = new Student();
```

- `Student` → class type
- `s1` → reference variable
- `new Student()` → object created in heap

Memory Perspective

- Reference variable stored in **Stack**
- Object stored in **Heap**
- Reference holds address of heap object

Key Interview Line

Java variables store references, not objects.

Assigning Values Using Reference

```
Student s1 = new Student();
s1.id = 101;
s1.name = "Arun";
```

- Data belongs to the object
- Method or variable accessed via reference

Reference Assignment

Code Example:

```
Student s1 = new Student();
Student s2 = s1;

s2.id = 50;
System.out.println(s1.id);
```

Output: `50`

Why?

Both references point to the same object.

Multiple Objects from Same Class

```
Student s1 = new Student();
Student s2 = new Student();

s1.id = 10;
s2.id = 20;
```

- Two different objects
- Same class, different memory
- Changes are independent

Null Reference

```
Student s1 = null;
s1.id = 5;    // Runtime error
```

Exception

`NullPointerException`

- Reference exists
- Object does not

Object Becoming Unreachable

```
Student s1 = new Student();
s1 = null;
```

- Object still exists in heap
- No reference points to it
- Eligible for Garbage Collection

Array of Objects – Concept

Definition

An array of objects is an array of reference variables, where each reference can point to an object.

```
Student[] arr = new Student[3];
```

- Creates array of 3 references
- NO Student objects created
- All elements initialized to `null`

Array of Objects

Declaration:

```
Student[] arr = new Student[3];
```

Object Creation:

```
arr[0] = new Student();
arr[1] = new Student();
arr[2] = new Student();
```

Important

Array stores references, not objects.

Memory Behavior of Array of Objects

```
Student[] arr = new Student[3];
System.out.println(arr[0]);
```

Output:

```
null
```

Key Point

Creating an array of objects does NOT create objects.

Common Runtime Error

```
Student[] arr = new Student[2];
arr[0].id = 10;
```

Runtime Exception

NullPointerException

- arr[0] is null
- Object not yet created

Creating Objects Using Loop

```
Student[] arr = new Student[3];

for(int i = 0; i < arr.length; i++) {
 arr[i] = new Student();
 arr[i].id = i + 1;
}
```

- Scalable approach
- Industry-preferred style

Accessing Array of Objects

Using Enhanced For Loop:

```
for (int i = 0; i < arr.length; i++) {
 System.out.println(arr[i].id);
}

for (Student s : arr) {
 System.out.println(s.id);
}

int i = 0;
while (i < arr.length) {
 System.out.println(arr[i].id);
 i++;
}
```

Passing Array of Objects to Method

```
for (int i = 0; i < arr.length; i++) {
 System.out.println(arr[i].id);
}

for (Student s : arr) {
 System.out.println(s.id);
}

int i = 0;
while (i < arr.length) {
 System.out.println(arr[i].id);
 i++;
}
```

Array of Objects vs Primitive Array

- Primitive array stores values directly
- Object array stores references
- Primitive arrays never store `null`
- Object arrays default to `null`

Common Misconceptions

- Declaring reference creates object
- Copying reference copies object
- Object has a name
- Java supports object copying by assignment

Brushing Up

Q1: Predict Output

```
Student s1 = new Student();
Student s2 = s1;
s1.id = 40;
System.out.println(s2.id);
```

Q2: Create two objects and prove they are independent.

Brushing Up

- Write a program using array of objects
- Assign values using loop
- Display all object details

Brushing Up

Conceptual Questions:

- Can an object exist without reference?
- Can a reference exist without object?
- Why does Java not support pointer arithmetic?

Scenario: Explain a real-time bug caused by null reference.

Brushing Up

1. Does `new Student[5]` create 5 objects? Why?
2. Why are array elements initialized to `null`?
3. Difference between:

```
Student[] arr = new Student[3];
Student s = new Student();
```

4. Why does accessing arr[0] without object creation fail?

Brushing Up

Predict Output:

```
Student[] arr = new Student[2];
arr[0] = new Student();
arr[0].id = 10;

Student temp = arr[0];
temp.id = 25;

System.out.println(arr[0].id);
```

Brushing Up

- Create array of 5 Student objects
- Assign id and name using loop
- Print details using enhanced for loop
- Explain memory allocation of array of objects
- Why is NullPointerException common in object arrays?

Thank You!

Stay Connected

Premanand S

Email: premanand.s@vit.ac.in
Phone: +91-7358679961

LinkedIn: linkedin.com/in/premsanand
Instagram: instagram.com/premsanand
WhatsApp Channel: anandsDataX

Google Scholar: Google Scholar Profile
GitHub: github.com/anandprems