Practical 1

Full Name:	Shivam Ravindra Khade
DIV:	IT-SEDA
Roll No.:	46
PRN:	1252130036
Subject:	OOP
Date of Conduction:	21-01-2026
Date of Submission:	06-02-2026

Software Used: Notepad, Windows Command Prompt,JDK

Problem Statement:

1. Implement Student class using following Concepts
    - All types of Constructors
    - Static variables and instance variables
    - Static blocks and instance blocks
    - Static methods and instance methods

Theory:

- **Constructor:-**
  A constructor is a special member of a class that is automatically called when an object is created to initialize the object's data.
- **Types of Constructors (commonly in Java)**

1. **Default Constructor** ○ Has
   no parameters.

- Initializes objects with default values.

2. **Parameterized Constructor** ○ Takes parameters. ○ Used to initialize objects with custom values.
3. **Copy Constructor** *(conceptual in Java)* ○ Creates a new object by copying another object's values. ○ Implemented using a parameterized constructor that takes an object of the same class.

➢ **Static Variables**

- Variables declared using static keyword.
- Shared by all objects of the class (single copy).
- Memory is allocated once at class loading time.

➢ **Instance Variables**

- Variables declared inside a class but outside methods, without static.
- Each object has its own copy.
- Memory is allocated when the object is created.

➢ **Static Block**

- A block of code declared using the static keyword.
- Executes once when the class is loaded. □ Used to initialize static variables.

➢ **Instance Block**

- A block of code without static keyword.
- Executes every time an object is created, before the constructor. □ Used to initialize instance variables.

➢ **Static Methods**

- Methods declared using the static keyword.
- Belong to the class, not to objects.
- Can be called without creating an object. □ Can directly access only static data.

➢ **Instance Methods**

- Methods declared without static.
- Belong to an object of the class.
- Must be called using an object reference.
- Can access both static and instance data.

Algorithm:

1) Start
2) When the program runs, the static block executes once and initializes name to "Unknown".

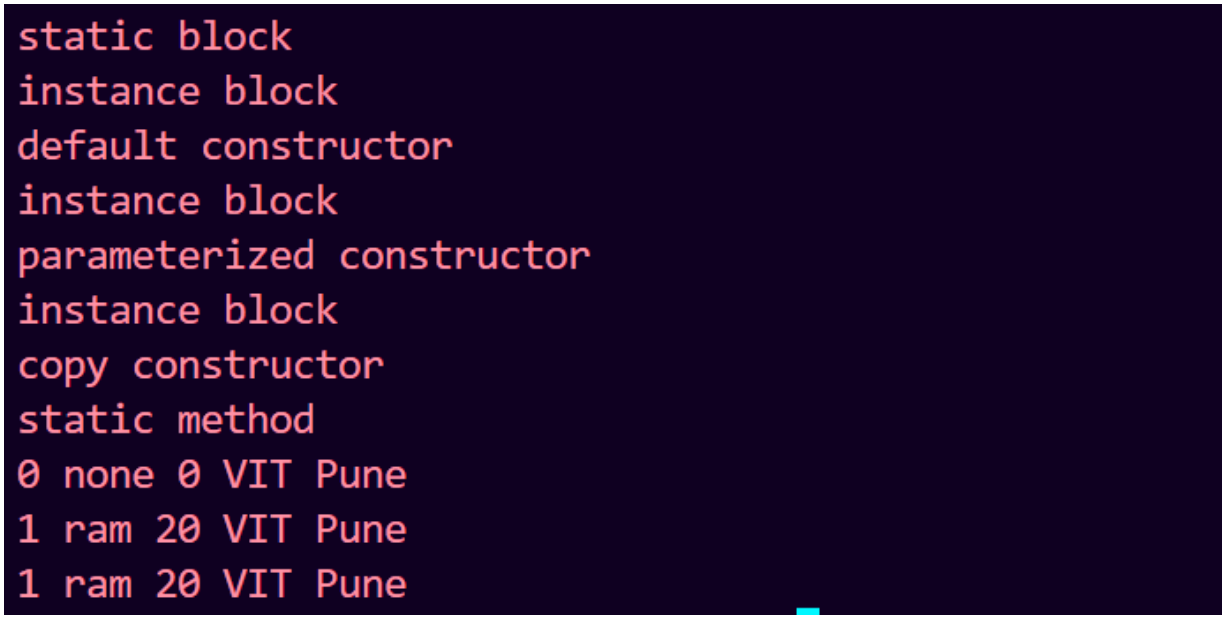
3) Create obj1 using the default constructor:
4) Instance block runs and sets RollNo = 44.
5) Default constructor prints roll number, name, and message.
6) Create obj2 using the parameterized constructor:
7) Instance block runs.
8) Name is set to "Sourabh" and printed.
9) Create obj3 using the copy constructor:
10) Instance block runs.
11) Name is copied from obj2.
12) Call UpdateName("Ram") using obj1:
13) Static variable name is updated to "Ram".
14) Stop

Code:

```
class Student {

    static String a;

    int b;
    String c;
    int d;

    static {
        a = "VIT Pune";
        System.out.println("static block");
    }

    {
        System.out.println("instance block");
    }

    Student() {
        b = 0;
        c = "none";
        d = 0;
        System.out.println("default constructor");
    }

    Student(int x, String y, int z) {
        b = x;
        c = y;
        d = z;
        System.out.println("parameterized constructor");
    }
```

```
    Student(Student s) {
        b = s.b;
        c = s.c;
        d = s.d;
        System.out.println("copy constructor");
    }

    static void m1() {
        System.out.println("static method");
    }

    void m2() {
        System.out.println(b + " " + c + " " + d + " " + a);
    }

    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student(1, "ram", 20);
        Student s3 = new Student(s2);

        Student.m1();
        s1.m2();
        s2.m2();
        s3.m2();
    }
}
```

Input/Output:



Conclusion:

This program demonstrates the execution order in Java. The static block runs only once when the class is loaded, while the instance block runs every time an object is created. Constructors (default, parameterized, and copy) initialize object data, and updating a static variable affects all objects of the class. Overall, the program clearly explains the behaviour of static members, instance members, and constructors in Java.