***Experiment. No. 01***

# **Title:** Program on classes and objects.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Excellent**  **4** | **Good**  **3** | **Satisfactory 2** | **Needs Improvement 1** | **Marks Obtained** |
| **Program Implementation** | Error-free, efficient code | Minor errors, good logic | Code works with some issues | Major errors or incomplete |  |
| **Output & Documentation** | Complete output with proper documentation | Good output, adequate documentation | Basic output shown | Incomplete or missing output |  |
| **Viva/Questions** | Excellent responses to all questions | Good responses to most questions | Satisfactory responses | Poor or no responses |  |
| **GitHub** | Properly Updated | No Proper organization | Incomplete | Not Updated |  |
| **Total** |  |  |  |  |  |

***Name: Salmani Asad Anwarul***

***UIN: 241A010***

***Roll No: 45***

***Date: 09/09/2025***

## Subject In-charge Sign:…………….

**Experiment No. 1**

**Aim: To write a Java program to demonstrate the concept of classes and objects, showcasing object-oriented programming principles such as encapsulation and reuse of code.**

**Software: IDE:** VS Code/Notepad

**Java Version:** Java jdk 21

**Operating System:** Windows/Linux/macOS

**Pre-lab Questions:**

**Q1. What is a class in Java?**

A class in Java is a blueprint or template used to create objects. It defines variables (data members) and methods (functions) that represent the behavior and state of the object.

**Q2. What is an object?**

An object is an instance of a class. It occupies memory and has its own identity, state, and behavior. Objects are created using the new keyword.

**Q3. What is the difference between a constructor and a method?**

* A **constructor** is a special method used to initialize objects. It has the same name as the class and no return type.
* A **method** defines behavior and can have any name and return type. Unlike constructors, methods must be explicitly called.

**Q4. What is the use of the new keyword?**

The **new** keyword is used to create objects dynamically. It allocates memory for the object and calls the constructor to initialize it.

### Theory:

### What is a Class?

### A class in Java is a blueprint or template from which objects are created.

### It defines data members (variables) and methods (functions) that represent the behavior of the

### object.

### Example:-

### class Student {

### int rollNo;

### String name;

### void display() {

### System.out.println("Roll No: " + rollNo + ", Name: " + name);

### }

### }

### What is an Object?

### An object is an instance of a class.

### It occupies memory and has its own identity, state, and behavior.

### Example:-

### Student s1 = new Student(); // s1 is an object of Student class

### What is a Constructor?

### A constructor is a special method in Java that is used to initialize objects of a class.

### It is automatically called when an object is created using the new keyword.

### A constructor has the same name as the class and does not have a return type, not even void.

### What is the new keyword in Java?

### The new keyword in Java is used to create objects dynamically during runtime. When you use

### new, it:

### Allocates memory in the heap for the object.Calls the constructor to initialize the object.

### Steps in Object-Oriented Programming: -

### 1. Define the class with variables and methods.

### 2. Create objects using the new keyword.

### 3. Access members using the object (objectName.variable or objectName.method()).

### 4. Initialize data using constructors or setter methods.

### 5. Retrieve or display data using getter methods or print statements.

### Problem Statement:

### To write a Java program to demonstrate the concept of classes and objects, showcasing object-oriented programming principles such as encapsulation and reuse of code.

### Program:

### public class Class1

### {

### int n;

### String s;

### public Class1()

### {

### System.out.println("Default Constructor");

### System.out.println("n="+n+"s="+s);

### }

### public void setN(int n1)

### {

### n=n1;

### }

### public void setS(String s1)

### {

### s=s1;

### }

### public int getN()

### {

### return n;

### }

### public String getS()

### {

### return s;

### }

### public static void main(String arg[])

### {

### Class1 c1=new Class1();

### c1.setN(12);

### c1.setS("Asad");

### System.out.println("n="+c1.getN());

### System.out.println("s="+c1.getS());

### }

### }

### Output:

### Post-lab Question:

**Q1. Can we have multiple classes in a single .java file?**

Yes, you can have multiple classes in one **.java** file. However, only one class can be declared **public**, and the filename must match the name of that public class.

**Q2. What happens if the filename doesn't match the public class name?**

The Java compiler will throw an error. The filename must exactly match the name of the public class to compile successfully**.**

**Q3. What is the role of the main() method in Java?**

The **main()** method is the entry point of any Java application. It is where the program starts execution:

**public static void main(String[] args)**

**Q4. What is the difference between Procedure Oriented Programming and Object-Oriented Programming?**

* **Procedure-Oriented Programming (POP):** Focuses on functions and procedures. Data is typically global and shared.
* **Object-Oriented Programming (OOP):** Focuses on objects that encapsulate both data and behavior. Promotes modularity, reusability, and abstraction.

**Lab Outcome:**

Understand and apply the fundamentals of Java Programming and Object-Oriented Programming.

**Conclusion:**

The above experiment demonstrates the use of classes and objects in Java. By defining a class Student, we encapsulate data (properties) and behavior (methods) in one reusable structure. Objects created by this class hold unique data but share common behaviors. This reflects core principles of object-oriented programming like modularity, reusability, and abstraction.