Experiment No. 3
Implement a program that demonstrates the concepts of class and objects
Date of Performance:
Date of Submission:

Aim: Implement a program that demonstrates the concepts of class and objects

Objective: To develop the ability of converting real time entity into objects and create their classes.

Theory:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties i.e., members and methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. Modifiers: A class can be public or has default access.
2. class keyword: class keyword is used to create a class.
3. Class name: The name should begin with a initial letter (capitalized by convention).
4. Superclass (if any): The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
5. Interfaces (if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
6. Body: The class body surrounded by braces, {}.

An OBJECT is a basic unit of Object-Oriented Programming and represents the real-life entities. A typical Java program creates many objects, which interact by invoking methods. An object consists of:

1. State: It is represented by attributes of an object. It also reflects the properties of an object.
2. Behavior: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. Identity: It gives a unique name to an object and enables one object to interact with other objects.

Code:

1}

1.Class

```java
class Student
 {
 float id;
 String name;
 public static void main(String args[])
 {
 Student s1=new Student();
 System.out.println(s1.id);
 System.out.println(s1.name);
 }
 }
```

2.Object

```java
class Student{
int id;
String name;
}
class Student2{
public static void main(String args[]){

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

 s1.id=101;
 s1.name="Ali";
```

```
        s2.id=102;

        s2.name="Amit";


        System.out.println(s1.id+" "+s1.name);

        System.out.println(s2.id+" "+s2.name);

        }

        }
```

Conclusion:

1) Comment on how you create a class template and their objects.

Creating a Class Template:

A class serves as a blueprint or template for creating objects. It defines the attributes (fields) and behaviors (methods) that objects of the class will have. In Java, you define a class using the class keyword. the class template is created using the class keyword, and you define the structure of the class by specifying its attributes (variables) and methods (if any). The class template acts as a blueprint for objects of that class.

Creating Objects:

Objects are instances of a class. They are created based on the class template. To create an object, you use the new keyword followed by the class name, followed by parentheses. This process is called instantiation. create objects of the "Student" class using Student s1 = new Student(); and Student s2 = new Student();. These lines instantiate two objects of the "Student" class, and now you have two distinct instances with their own attributes and behaviors.