



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

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



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

**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}  
    class Rectangle{  
        int length;  
        int width;  
        void insert(int l, int w){  
            length=l;  
            width=w;  
        }  
        void calculateArea(){System.out.println(length*width);}  
    }  
    class TestRectangle1 {  
        public static void main(String args[]){  
            Rectangle r1=new Rectangle();  
            Rectangle r2=new Rectangle();  
            r1.insert(7,9);  
            r2.insert(5,12);  
            r1.calculateArea();  
            r2.calculateArea();  
        }  
    }
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Users\HP\OneDrive\Desktop\Charmi
C:\Users\HP\OneDrive\Desktop\Charmi>javac Rectangle.java
C:\Users\HP\OneDrive\Desktop\Charmi>java Rectangle.java
63
68
C:\Users\HP\OneDrive\Desktop\Charmi>
```

### Conclusion:

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

Define a Class Template:

Class Definition:

Begin by using the class keyword, followed by a chosen class name, to lay the foundation for a class template. Inside this class, declare fields or attributes that will characterize the state of objects. Define constructors to establish the initial state of objects and introduce methods that encapsulate the behavior and actions to be performed by the objects.

Object Creation:

Objects are brought to life using the new keyword, coupled with the class constructor. These freshly created instances can be assigned to variables for further manipulation.

Accessing Attributes and Methods:

Employ the dot notation as a bridge to interact with the fields and invoke the methods associated with the object. This allows you to both retrieve and influence the state and behavior of the objects you've instantiated.

