

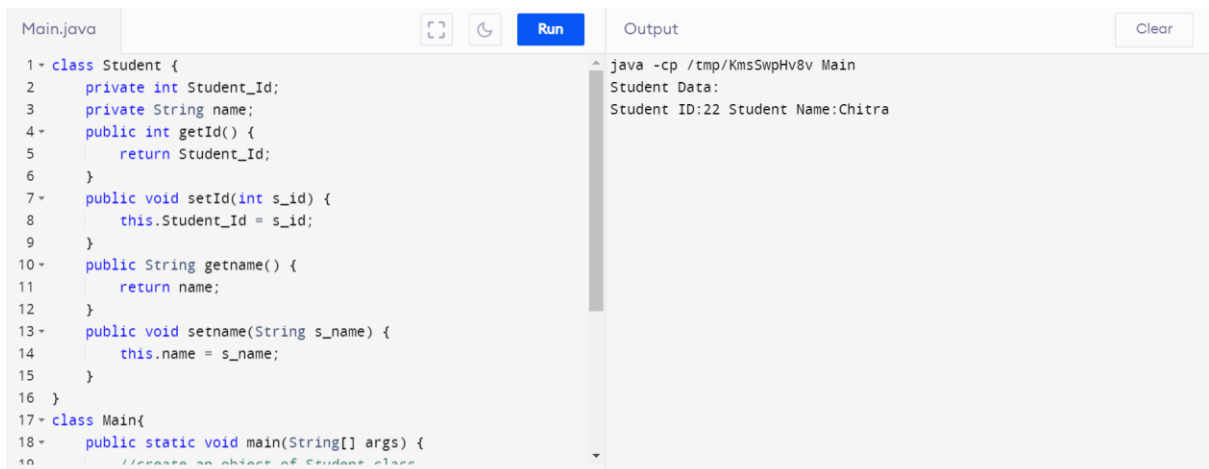
## JAVA PROGRAMS IN OOPS CONCEPTS

### ENCAPSULATION

```
class Student {
    private int Student_Id;
    private String name;
    public int getId() {
        return Student_Id;
    }
    public void setId(int s_id) {
        this.Student_Id = s_id;
    }
    public String getname() {
        return name;
    }
    public void setname(String s_name) {
        this.name = s_name;
    }
}

class Main{
    public static void main(String[] args) {
        //create an object of Student class
        Student s=new Student();
        //set fields values using setter methods
        s.setId (22);
        s.setname("Chitra");
        //print values using getter methods
        System.out.println("Student Data:" + "\nStudent ID:" + s.getId()
            + " Student Name:" + s.getname());
    }
}
```

### OUTPUT



The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1- class Student {
2-     private int Student_Id;
3-     private String name;
4-     public int getId() {
5-         return Student_Id;
6-     }
7-     public void setId(int s_id) {
8-         this.Student_Id = s_id;
9-     }
10-    public String getname() {
11-        return name;
12-    }
13-    public void setname(String s_name) {
14-        this.name = s_name;
15-    }
16- }
17- class Main{
18-     public static void main(String[] args) {
19-         //create an object of Student class
```

The output window on the right shows the command executed: `java -cp /tmp/KmsSwpHv8v Main`. The output is:

```
Student Data:
Student ID:22 Student Name:Chitra
```

## 2.ABSTRACT METHOD

```

abstract class Bank{
    abstract int getInterestRate();
}

class Citi extends Bank{
    int getInterestRate(){return 7;}
}

class HDFC extends Bank{
    int getInterestRate(){return 6;}
}

class Main{
    public static void main(String args[]){
        Bank b;

        b = new Citi ();

        System.out.println("Citi Rate of Interest is: "+b.getInterestRate()+"%");

        b = new HDFC ();

        System.out.println("HDFC Rate of Interest is: "+b.getInterestRate()+"%");
    }
}

```

## OUTPUT

```

Main.java
1- abstract class Bank{
2-     abstract int getInterestRate();
3- }
4
5- class Citi extends Bank{
6-     int getInterestRate(){return 7;}
7- }
8
9- class HDFC extends Bank{
10-     int getInterestRate(){return 6;}
11- }
12
13- class Main{
14-     public static void main(String args[]){
15-         Bank b;
16-         b = new Citi ();
17-         System.out.println("Citi Rate of Interest is: "+b
18-             .getInterestRate()+"%");
19-         b = new HDFC ();
20-         System.out.println("HDFC Rate of Interest is: "+b
21-             .getInterestRate()+"%");
22-     }
23- }

```

```

Output
java -cp /tmp/KmsSwpHV8v Main
Citi Rate of Interest is: 7%HDFC Rate of Interest is: 6%

```

## 3.INHERITANCE

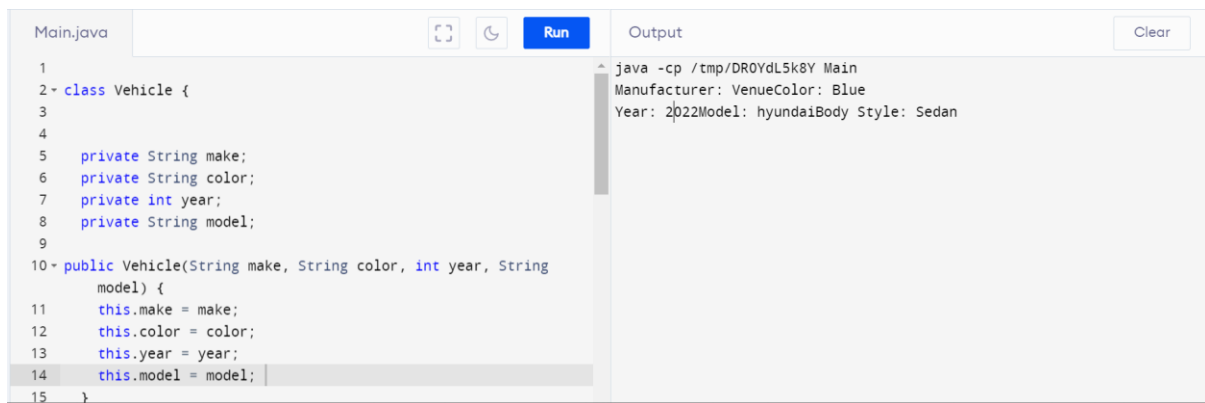
```

class Vehicle
private String make;
private String color;
private int year;
private String model;
public Vehicle(String make, String color, int year, String model) {
    this.make = make;
    this.color = color;
    this.year = year;
    this.model = model;
} public void printDetails() {
    System.out.println("Manufacturer: " + make);
    System.out.println("Color: " + color);
    System.out.println("Year: " + year);
    System.out.println("Model: " + model);
}
}

class Car extends Vehicle {
private String bodyStyle;
public Car(String make, String color, int year, String model, String bodyStyle) {
    super(make, color, year, model); //calling parent class constructor
    this.bodyStyle = bodyStyle;
} public void carDetails() {
    printDetails();
    System.out.println("Body Style: " + bodyStyle);
}
}class Main {
public static void main(String[] args) {
    Car elantraSedan = new Car("Venue", "Blue", 2022, "hyundai", "Sedan");
    elantraSedan.carDetails();
}
}

```

## OUTPUT



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a 'Vehicle' class with private attributes 'make', 'color', 'year', and 'model', and a public constructor. The output window shows the result of running the program: 'java -cp /tmp/DR0YdL5k8Y Main', 'Manufacturer: VenueColor: Blue', 'Year: 2022Model: hyundaiBody Style: Sedan'.

```
1  
2- class Vehicle {  
3  
4  
5   private String make;  
6   private String color;  
7   private int year;  
8   private String model;  
9  
10- public Vehicle(String make, String color, int year, String  
    model) {  
11   this.make = make;  
12   this.color = color;  
13   this.year = year;  
14   this.model = model;  
15 }
```

Output: java -cp /tmp/DR0YdL5k8Y Main  
Manufacturer: VenueColor: Blue  
Year: 2022Model: hyundaiBody Style: Sedan

## 4.RUNTIME POLYMORPHISM

```
class Animal{  
  
    void eat(){System.out.println("animal is eating...");  
  
    }  
  
}  
  
class Dog extends Animal{  
  
    void eat(){System.out.println("dog is eating...");  
  
    }  
  
}  
  
class BabyDog1 extends Dog{  
  
    public static void main(String args[]){  
  
        Animal a=new BabyDog1();  
  
        a.eat();  
  
    }  
  
}
```

## OUTPUT



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines three classes: 'Animal' with an 'eat()' method, 'Dog' extending 'Animal' and overriding 'eat()', and 'BabyDog1' extending 'Dog'. The 'main' method in 'BabyDog1' creates an 'Animal' object and calls 'eat()'. The output window shows the result of running the program: 'java -cp /tmp/6sKigesXwj BabyDog1', 'dog is eating...'.

```
1- class Animal{  
2   void eat(){System.out.println("animal is eating...");  
3   }  
4 }  
5- class Dog extends Animal{  
6   void eat(){System.out.println("dog is eating...");  
7   }  
8 }  
9- class BabyDog1 extends Dog{  
10- public static void main(String args[]){  
11   Animal a=new BabyDog1();  
12   a.eat();  
13 }
```

Output: java -cp /tmp/6sKigesXwj BabyDog1  
dog is eating...