**CDAC Mumbai PG-DAC August 24**

**Assignment No- 5**

1. Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.

Code:

public class Bank {

    protected double balance;

    public void deposit(double amount) {

        balance += amount;

        System.out.println("Deposited: " + amount + ". Current balance: " + balance);

    }

    public void withdraw(double amount) {

        if (amount <= balance) {

            balance -= amount;

        }

        else{

            System.out.println("Insufficient Balaence : ");

        }

    }

}

class savingsAccoutns extends Bank{

    private double withdrawlimit = 10000;

    public void withdraw(double amount){

        if(amount > withdrawlimit){

            System.out.println("Withdraw Limits exceeds.... ");

        }

        else{

            super.withdraw(amount);

            System.out.println("Withdrew: " + amount + ". Remaining balance: " + balance);

        }

    }

    public static void main(String args[]) {

        savingsAccoutns mySavings = new savingsAccoutns();

        mySavings.deposit(25000);

        mySavings.withdraw(3000);

        mySavings.withdraw(5000);

        mySavings.withdraw(1000);

    }

}

A screenshot of a computer

Description automatically generated

1. Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.

Code:

package org.assign6;

public class Assgnment {

    // Base class Vehicle

    static class Vehicle {

        protected String make;

        protected int year;

        // Constructor to initialize make and year

        public Vehicle(String make, int year) {

            this.make = make;

            this.year = year;

        }

        // Method to display vehicle details

        public void displayDetails() {

            System.out.println("Vehicle Make: " + make);

            System.out.println("Vehicle Year: " + year);

        }

    }

    // Derived class Car that extends Vehicle

    static class Car extends Vehicle {

        private String model;

        // Constructor to initialize make, year, and model

        public Car(String make, int year, String model) {

            super(make, year);  // Call Vehicle's constructor

            this.model = model; // Initialize model

        }

        // Overriding the displayDetails method to include model information

        @Override

        public void displayDetails() {

            super.displayDetails();  // Call the parent class method

            System.out.println("Car Model: " + model);  // Display the model of the car

        }

    }

    // Main method to test the functionality

    public static void main(String[] args) {

        // Create a Car object

        Car myCar = new Car("Toyota", 2020, "Corolla");

        // Display car details

        myCar.displayDetails();

    }

}

A screenshot of a computer

Description automatically generated

1. Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

Code:

package org.assign6;

public class Assignment2 {

    static class Animal {

        protected String name;

        public Animal(String name) {

            this.name = name;

        }

        public void eat() {

            System.out.println(name + " is eating.");

        }

        public void sleep() {

            System.out.println(name + " is sleeping.");

        }

    }

    static class Dog extends Animal {

        public Dog(String name) {

            super(name);

        }

        public void bark() {

            System.out.println(name + " is barking.");

        }

    }

    public static void main(String[] args) {

        Animal animal = new Animal("Generic Animal");

        animal.eat();

        animal.sleep();

        Dog dog = new Dog("Buddy");

        dog.eat();

        dog.sleep();

        dog.bark();

    }

}

A screenshot of a computer

Description automatically generated

1. Build a class Student which contains details about the Student and compile and run its

instance.

Code:

package org.assign6;

public class Student {

    private String name;

    private int roll\_no;

    private char grade;

    // Constructor to initialize the attributes

    public Student(String name, int roll\_no, char grade) {

        this.name = name;

        this.roll\_no = roll\_no;

        this.grade = grade;

    }

    // Getter for name

    public String getName() {

        return name;

    }

    // Getter for roll number

    public int getRollNumber() {

        return roll\_no;

    }

    // Getter for grade

    public char getGrade() {

        return grade;

    }

    // Method to display the student's details

    public void displayDetails() {

        System.out.println("Student Name: " + name);

        System.out.println("Roll Number: " + roll\_no);

        System.out.println("Grade: " + grade);

    }

    // Main method to test the functionality

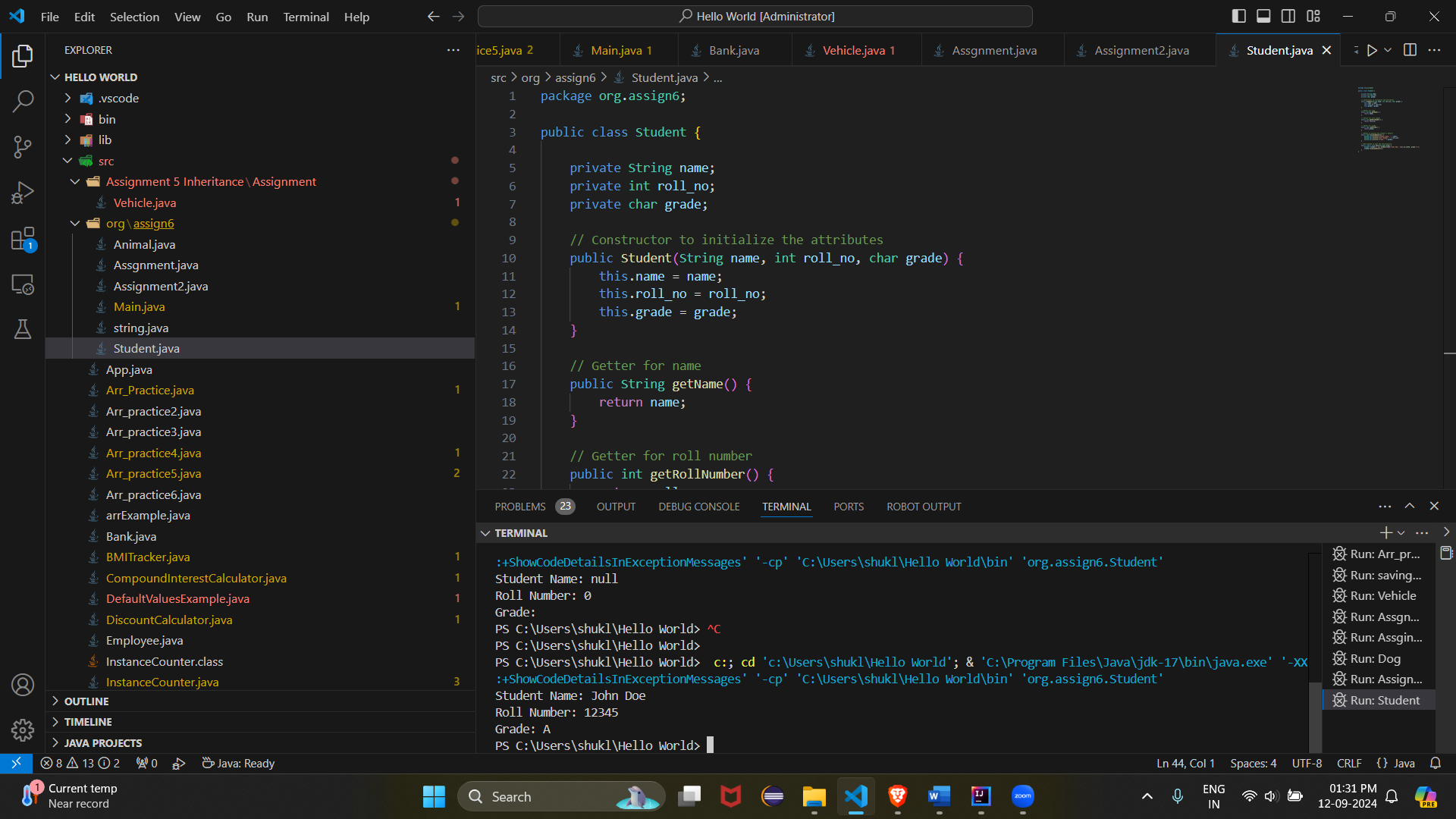
    public static void main(String[] args) {

        Student student = new Student("John Doe", 12345, 'A');

        student.displayDetails();

    }

}



1. Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.

Code: package org.assign6;

public class Assignment5 {

    // Base class

    static class Vehicle {

        public void startEngine() {

            System.out.println("Starting the vehicle engine...");

        }

        public void stopEngine() {

            System.out.println("Stopping the vehicle engine...");

        }

    }

    // Subclass Car

    static class Car extends Vehicle {

        @Override

        public void startEngine() {

            System.out.println("Starting the car engine with a key...");

        }

        @Override

        public void stopEngine() {

            System.out.println("Turning off the car engine...");

        }

    }

    // Subclass Motorcycle

    static class Motorcycle extends Vehicle {

        @Override

        public void startEngine() {

            System.out.println("Starting the motorcycle engine with a button...");

        }

        @Override

        public void stopEngine() {

            System.out.println("Turning off the motorcycle engine...");

        }

    }

    // Main class to test the functionality

    public static void main(String[] args) {

        Vehicle myCar = new Car();

        Vehicle myMotorcycle = new Motorcycle();

        System.out.println("Car actions:");

        myCar.startEngine();

        myCar.stopEngine();

        System.out.println("\nMotorcycle actions:");

        myMotorcycle.startEngine();

        myMotorcycle.stopEngine();

    }

}

A screenshot of a computer program

Description automatically generated