**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:

**package** Q1;

**public** **class** BankAccount {

**protected** **double** balance;

**public** BankAccount(**double** balance) {

**this**.balance = balance;

}

**public** **void** deposit(**double** amount) {

balance += amount;

System.***out***.println("Deposited: ₹" + amount + ", New Balance: ₹" + balance);

}

**public** **void** withdraw(**double** amount) {

**if** (balance >= amount) {

balance -= amount;

System.***out***.println("Withdrawn: ₹" + amount + ", New Balance: ₹" + balance);

} **else** {

System.***out***.println("Insufficient funds. Withdrawal failed.");

}

}

**public** **double** getBalance() {

**return** balance;

}

}

**class** SavingsAccount **extends** BankAccount {

**private** **double** withdrawalLimit;

**public** SavingsAccount(**double** balance, **double** withdrawalLimit) {

**super**(balance);

**this**.withdrawalLimit = withdrawalLimit;

}

@Override

**public** **void** withdraw(**double** amount) {

**if** (amount > withdrawalLimit) {

System.***out***.println("Withdrawal limit exceeded. Limit: ₹" + withdrawalLimit);

} **else** {

**super**.withdraw(amount);

}

}

}

**package** Q1;

**public** **class** Text {

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

SavingsAccount savingsAccount = **new** SavingsAccount(15000, 5000);

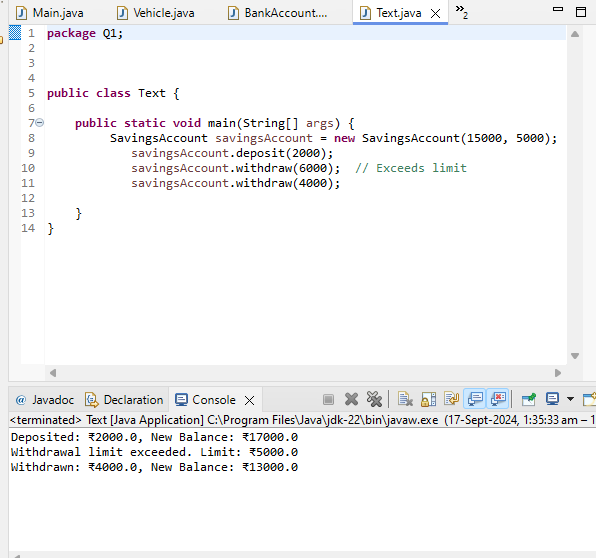
savingsAccount.deposit(2000);

savingsAccount.withdraw(6000); // Exceeds limit

savingsAccount.withdraw(4000);

}

}



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** Q2;

**public** **class** Vehicle {

**private** String make;

**private** **int** year;

**public** Vehicle(String make, **int** year) {

**this**.make = make;

**this**.year = year;

}

**public** String getDetails() {

**return** "Make: " + make + ", Year: " + year;

}

}

**class** Car **extends** Vehicle {

**private** String model;

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

**super**(make, year);

**this**.model = model;

}

@Override

**public** String getDetails() {

**return** **super**.getDetails() + ", Model: " + model;

}

}

**package** Q2;

**public** **class** Main {

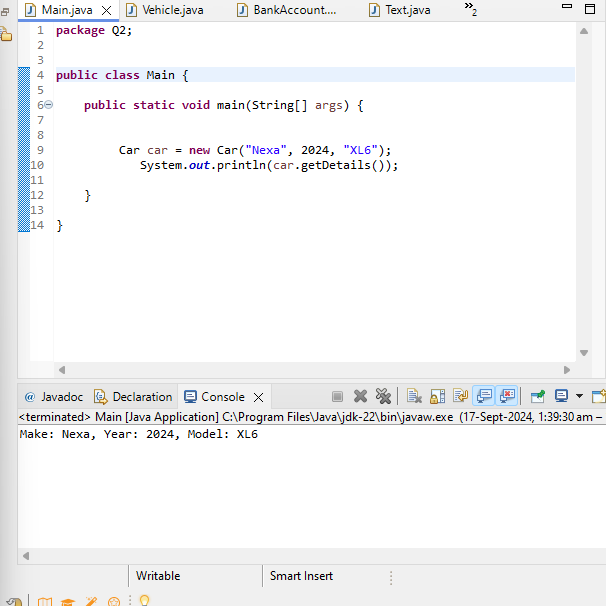
**public** **static** **void** main(String[] args) {

Car car = **new** Car("Nexa", 2024, "XL6");

System.***out***.println(car.getDetails());

}

}



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.

**package** Q3;

**public** **class** Animal {

String name;

**public** Animal(String name) {

**this**.name = name;

}

**public** **void** eat() {

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

}

**public** **void** sleep() {

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

}

}

**class** Dog **extends** Animal{

**public** Dog(String name) {

**super**(name);

// **TODO** Auto-generated constructor stub

}

**public** **void** bark() {

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

}

}

**package** Q3;

**public** **class** Test {

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

Animal anim = **new** Animal("Cat");

anim.eat();

anim.sleep();

Dog dog = **new** Dog("Lucy");

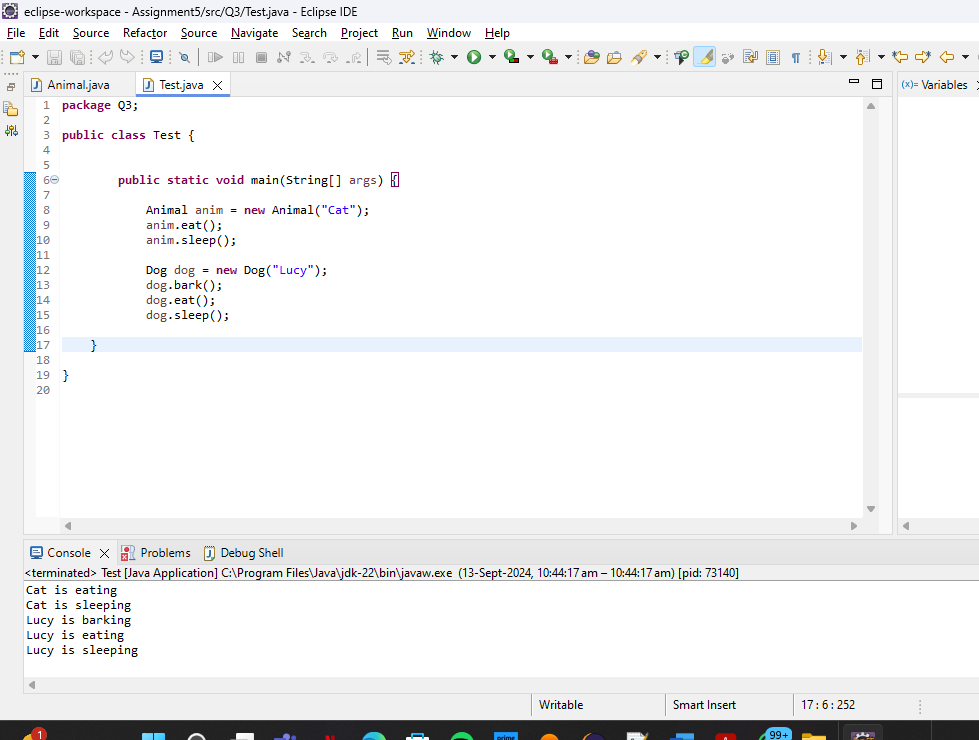
dog.bark();

dog.eat();

dog.sleep();

}

}



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

instance.

CODE:

**package** Q4;

**public** **class** Students {

String name;

**int** rollNumber;

**public** Students(String name, **int** rollNumber) {

**this**.name = name;

**this**.rollNumber = rollNumber;

}

**public** **void** printRecord() {

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

System.***out***.println("Roll Number: "+**this**.rollNumber);

}

}

**package** Q4;

**public** **class** Test {

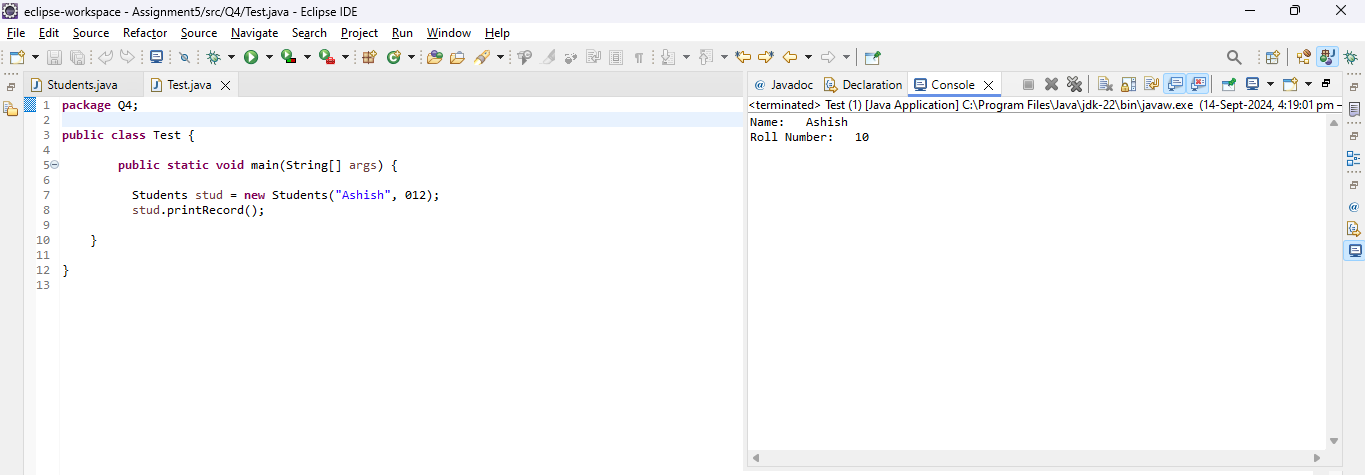
**public** **static** **void** main(String[] args) {

Students stud = **new** Students("Ashish", 012);

stud.printRecord();

}

}



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** Q5;

**public** **class** Vehicle {

**public** **void** startEngine() {

System.***out***.println("Vehicle Engine stsrts.");

}

**public** **void** stopEngine() {

System.***out***.println("Vehicle Engine Stops.");

}

}

**class** Car **extends** Vehicle{

@Override

**public** **void** startEngine() {

System.***out***.println("Car Engine Starts with button.");

}

@Override

**public** **void** stopEngine() {

System.***out***.println("Car Engine stops with button.");

}

}

**class** Motorcycle **extends** Vehicle{

@Override

**public** **void** startEngine() {

System.***out***.println("Motorcycle Engine Starts with button.");

}

@Override

**public** **void** stopEngine() {

System.***out***.println("Motorcycle Engine stops with button.");

}

}

**package** Q5;

**public** **class** TestVehicle {

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

Vehicle car = **new** Car();

car.startEngine();

car.stopEngine();

Vehicle bike = **new** Motorcycle();

bike.startEngine();

bike.stopEngine();

}

}

