**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.

**Program:**

class BankAccount {

private String accountHolderName;

private double balance;

public BankAccount(String accountHolderName, double balance) {

this.accountHolderName = accountHolderName;

this.balance = balance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println(amount + " deposited. New balance: " + balance);

} else {

System.out.println("Invalid deposit amount.");

}

}

public boolean withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

System.out.println(amount + " withdrawn. New balance: " + balance);

return true;

} else {

System.out.println("Insufficient balance or invalid amount.");

return false;

}

}

public double getBalance() {

return balance;

}

public String getAccountHolderName() {

return accountHolderName;

}

}

class SavingsAccount extends BankAccount {

private double withdrawalLimit;

public SavingsAccount(String accountHolderName, double balance, double withdrawalLimit) {

super(accountHolderName, balance);

this.withdrawalLimit = withdrawalLimit;

}

@Override

public boolean withdraw(double amount) {

if (amount > withdrawalLimit) {

System.out.println("Withdrawal amount exceeds the limit of " + withdrawalLimit);

return false;

}

return super.withdraw(amount);

}

}

public class Main {

public static void main(String[] args) {

SavingsAccount savingsAccount = new SavingsAccount("Ashwini Patil", 5000.0, 2000.0);

System.out.println("Account Holder: " + savingsAccount.getAccountHolderName());

System.out.println("Initial Balance: " + savingsAccount.getBalance());

savingsAccount.deposit(1500.0);

System.out.println("\nAttempting to withdraw 2500:");

savingsAccount.withdraw(2500.0);

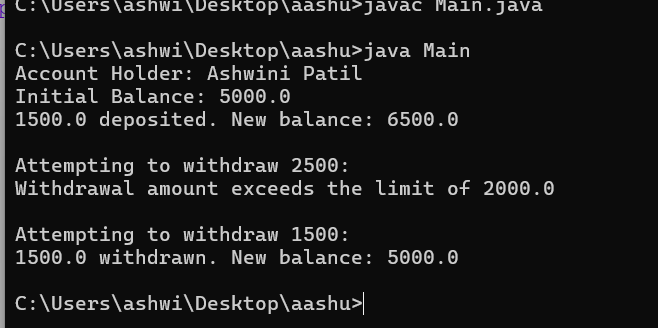
System.out.println("\nAttempting to withdraw 1500:");

savingsAccount.withdraw(1500.0);

}

}

**Output:**



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.

**Program:**

class Vehicle {

private String make;

private int year;

public Vehicle(String make, int year) {

this.make = make;

this.year = year;

}

public String getMake() {

return make;

}

public int getYear() {

return year;

}

}

class Car extends Vehicle {

private String model;

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

super(make, year);

this.model = model;

}

public String getModel() {

return model;

}

public void displayDetails() {

System.out.println("Make: " + getMake());

System.out.println("Year: " + getYear());

System.out.println("Model: " + getModel());

}

}

public class Main1 {

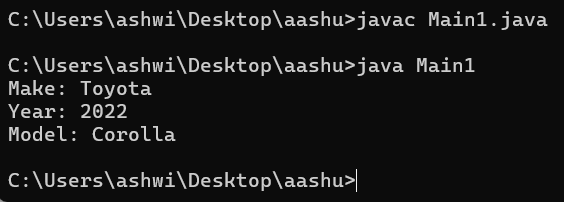
public static void main(String[] args) {

Car car = new Car("Toyota", 2022, "Corolla");

car.displayDetails();

}

}



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.

**Program:**

class Animal {

private 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.");

}

public String getName() {

return name;

}

}

class Dog extends Animal {

public Dog(String name) {

super(name);

}

public void bark() {

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

}

}

public class Main2 {

public static void main(String[] args) {

Animal animal = new Animal("Tiger");

animal.eat();

animal.sleep();

System.out.println();

Dog dog = new Dog("Buddy");

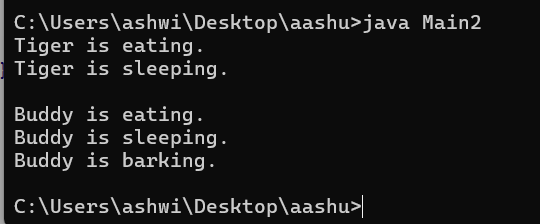
dog.eat();

dog.sleep();

dog.bark();

}

}



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

instance.

**Program:**

class Student {

private String name;

private int rollNumber;

private String grade;

public Student(String name, int rollNumber, String grade) {

this.name = name;

this.rollNumber = rollNumber;

this.grade = grade;

}

public String getName() {

return name;

}

public int getRollNumber() {

return rollNumber;

}

public String getGrade() {

return grade;

}

public void displayDetails() {

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

System.out.println("Roll Number: " + getRollNumber());

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

}

}

public class Main3 {

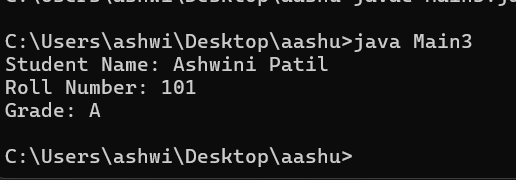
public static void main(String[] args) {

Student student = new Student("Ashwini Patil", 101, "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.

**Program:**

class Vehicle {

public void startEngine() {

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

}

public void stopEngine() {

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

}

}

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("Stopping the car engine by turning off the key.");

}

}

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("Stopping the motorcycle engine by pressing the button.");

}

}

public class Main4 {

public static void main(String[] args) {

Vehicle myCar = new Car();

myCar.startEngine();

myCar.stopEngine();

System.out.println();

Vehicle myMotorcycle = new Motorcycle();

myMotorcycle.startEngine();

myMotorcycle.stopEngine();

}

}

