**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.

**package** com.assign;

**import** java.util.Scanner;

**class** BankAccount {

**private** **double** balance;

**public** BankAccount(**double** initialBalance) {

**this**.balance = initialBalance;

}

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

**if** (amount > 0) {

balance += amount;

} **else** {

System.***out***.println("Deposit amount must be positive.");

}

}

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

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

balance -= amount;

} **else** {

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

}

}

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

**return** balance;

}

}

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

**private** **static** **final** **double** ***WITHDRAWAL\_LIMIT*** = 1000.0; // Example limit

// Constructor to initialize balance in base class

**public** SavingsAccount(**double** initialBalance) {

**super**(initialBalance);

}

// Override withdraw method to impose a limit on the withdrawal amount

@Override

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

**if** (amount > 0 && amount <= ***WITHDRAWAL\_LIMIT*** && amount <= getBalance()) {

**super**.withdraw(amount);

} **else** {

**if** (amount > ***WITHDRAWAL\_LIMIT***) {

System.***out***.println("Withdrawal amount exceeds the limit.");

} **else** {

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

}

}

}

}

**public** **class** Assign5\_q1 {

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

SavingsAccount myAccount = **new** SavingsAccount(1500.0);

myAccount.deposit(500.0);

System.***out***.println("Balance after deposit: " + myAccount.getBalance());

myAccount.withdraw(800.0);

System.***out***.println("Balance after withdrawal: " + myAccount.getBalance());

myAccount.withdraw(1200.0);

myAccount.withdraw(500.0);

}

}

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.

public 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;

}

public void displayDetails() {

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

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

}

}

public class Car extends Vehicle {

private String model;

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

super(make, year); // Call the constructor of the base class

this.model = model;

}

public String getModel() {

return model;

}

@Override

public void displayDetails() {

super.displayDetails(); // Call the displayDetails method of the base class

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

}

}

public class Main {

public static void main(String[] args) {

// Create a Car object with make, year, and model

Car myCar = new Car("Toyota", 2024, "Camry");

// Display details of the Car object

myCar.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.

public class Animal {

private String name;

public Animal(String name) {

this.name = name;

}

public String getName() {

return name;

}

public void eat() {

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

}

public void sleep() {

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

}

}

public class Dog extends Animal {

public Dog(String name) {

super(name);

}

public void bark() {

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

}

}

public class Main {

public static void main(String[] args) {

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

Dog myDog = new Dog("Buddy");

System.out.println("Animal:");

myAnimal.eat();

myAnimal.sleep();

System.out.println("\nDog:");

myDog.eat();

myDog.sleep();

myDog.bark();

}

}

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

instance.

public class Student {

private String name;

private int age;

private double marks;

public Student(String name, int age, double marks) {

this.name = name;

this.age = age;

this.marks = marks;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

public double getMarks() {

return marks;

}

public void displayDetails() {

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

System.out.println("Age: " + age);

System.out.println("Marks: " + marks);

}

}

public class Main {

public static void main(String[] args) {

Student student1 = new Student("Alice", 20, 89.5);

System.out.println("Student Details:");

student1.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.

public 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.");

}

}

public class Car extends Vehicle {

@Override

public void startEngine() {

System.out.println("Car engine is starting with a roar.");

}

@Override

public void stopEngine() {

System.out.println("Car engine is stopping with a smooth shutdown.");

}

}

public class Motorcycle extends Vehicle {

@Override

public void startEngine() {

System.out.println("Motorcycle engine is starting with a vroom.");

}

@Override

public void stopEngine() {

System.out.println("Motorcycle engine is stopping with a quick shutdown.");

}

}

public class Main {

public static void main(String[] args) {

Vehicle myCar = new Car();

myCar.startEngine();

myCar.stopEngine();

Vehicle myMotorcycle = new Motorcycle();

myMotorcycle.startEngine();

myMotorcycle.stopEngine();

}

}