**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.

**class** BankAccount {

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

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

**this**.balance = initialBalance;

}

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

**if** (amount > 0) {

balance += amount;

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

} **else** {

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

}

}

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

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

balance -= amount;

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

} **else** {

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

}

}

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

**return** balance;

}

}

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

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

**super**(initialBalance);

}

@Override

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

**if** (amount > 0 && getBalance() - amount >= 100) {

**super**.withdraw(amount);

} **else** {

System.out.println("Withdrawal denied. Minimum balance of INR100 required.");

}

}

}

**public** **class** question1 {

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

BankAccount account = **new** BankAccount(500);

account.deposit(200);

account.withdraw(100);

System.out.println("BankAccount balance: " + account.getBalance());

SavingsAccount savings = **new** SavingsAccount(500);

savings.deposit(200);

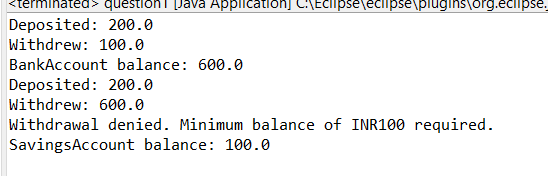
savings.withdraw(600);

savings.withdraw(100);

System.out.println("SavingsAccount balance: " + savings.getBalance());

}

}



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.

**class** Vehicle

{

**private** String sMake;

**private** **int** iYear;

**public** Vehicle(String sCompanyName,**int** iPeriod)

{

**this**.sMake=sCompanyName;

**this**.iYear=iPeriod;

}

**public** **void** displayDetails()

{

System.***out***.println(**this**.sMake);

System.***out***.println(**this**.iYear);

}

}

**class** car **extends** Vehicle

{

**private** String sModel;

**public** car(String sCompanyName,**int** iPeriod,String sCarName)

{

**super**(sCompanyName,iPeriod);

**this**.sModel=sCarName;

}

**public** **void** displayDetails()

{ **super**.displayDetails();

System.***out***.println(**this**.sModel);

}

}

**class** question2

{

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

{

car cobj=**new** car("honda",2024,"Civic");

cobj.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.

**class** Animal {

**public** 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.");

}

}

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

**public** Dog(String name) {

**super**(name);

}

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

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

}

}

**public** **class** question3 {

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

Animal aobj = **new** Animal("tiger"); //base class object- parameterised const.gets called

aobj.eat(); // base class obj calls method eat()

aobj.sleep(); // base class obj calls method sleep()

Dog dobj = **new** Dog("ollie"); // subclass obj created-parameterised cons.gets called

dobj.eat(); // subclass obj calls eat method

dobj.sleep(); // subclas obj calls sleeps methods

dobj.bark(); // subclss obj calls bark methods

}

}

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

instance.

**class** student

{

**private** String sName;

**private** **int** iAge;

**private** **float** fPercentage;

**private** String sCity;

**public** student(String sName,**int** iAge,**float** fPercentage,String sCity )

{

**this**.sName=sName;

**this**.iAge=iAge;

**this**.fPercentage=fPercentage;

**this**.sCity=sCity;

}

**public** String getName() {

**return** sName;

}

**public** **int** getAge() {

**return** iAge;

}

**public** **float** getPercentage() {

**return** fPercentage;

}

**public** **void** setName(String name) {

**this**.sName = sName;

}

**public** **void** setAge(**int** age) {

**this**.iAge = iAge;

}

**public** **void** setPercentage(**float** per) {

**this**.fPercentage = fPercentage;

}

**public** String getDetails() {

**return** "Name: " + sName + ", Age: " + iAge + ", Percentage: " + fPercentage;

}

}

**public** **class** question4

{

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

{

student sobj=**new** student("Pratik Jagtap",27,81.2f,"Nashik");

System.out.println(sobj.getDetails());

}

}

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.

**class** Vehicle

{

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

{

}

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

{

}

}

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

{

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

{

System.out.println("car's ignition is on");

}

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

{

System.out.println("car's ignition is off");

}

}

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

{

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

{

System.out.println("Motorcycle's key is turned on");

}

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

{

System.out.println("Motorcycle's key is turned off");

}

}

**class** question5

{

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

{

Car cobj=**new** Car();

Motorcycle mobj=**new** Motorcycle();

cobj.startEngine();

cobj.stopEngine();

mobj.startEngine();

mobj.stopEngine();

}

}