**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 code:**

**package Bank.org;**

**class BankAccount {**

**private double balance;**

**public BankAccount(double balance) {**

**this.balance = balance;**

**}**

**public double getBalance() {**

**return balance;**

**}**

**public void setBalance(double balance) {**

**this.balance = balance;**

**}**

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

**if (amount > 0) {**

**setBalance(getBalance() + amount);**

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

**} else {**

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

**}**

**}**

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

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

**setBalance(getBalance() - amount);**

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

**} else {**

**System.*out*.println("Invalid withdraw amount or insufficient balance.");**

**}**

**}**

**}**

**class SavingsAccount extends BankAccount {**

**private double withdrawLimit;**

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

**super(balance);**

**this.withdrawLimit = withdrawLimit;**

**}**

**public double getWithdrawLimit() {**

**return withdrawLimit;**

**}**

**public void setWithdrawLimit(double withdrawLimit) {**

**this.withdrawLimit = withdrawLimit;**

**}**

**@Override**

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

**if (amount > getWithdrawLimit()) {**

**System.*out*.println("Withdrawal amount exceeds the limit of: " + getWithdrawLimit());**

**} else {**

**super.withdraw(amount); // Call the withdraw method of the base class**

**}**

**}**

**}**

**public class Bank {**

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

**BankAccount bankAccount = new BankAccount(1000);**

**System.*out*.println("Initial BankAccount Balance: " + bankAccount.getBalance());**

**bankAccount.deposit(500);**

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

**bankAccount.withdraw(300);**

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

**SavingsAccount savingsAccount = new SavingsAccount(2000, 500);**

**System.*out*.println("\nInitial SavingsAccount Balance: " + savingsAccount.getBalance());**

**savingsAccount.deposit(1000);**

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

**savingsAccount.withdraw(600); // This should trigger the limit warning**

**System.*out*.println("Balance after attempting to withdraw 600: " + savingsAccount.getBalance());**

**savingsAccount.withdraw(400);**

**System.*out*.println("Balance after withdrawing 400: " + savingsAccount.getBalance());**

**}**

**}**

**Output:**

**Initial BankAccount Balance: 1000.0**

**Deposited: 500.0**

**Balance after deposit: 1500.0**

**Withdrew: 300.0**

**Balance after withdrawal: 1200.0**

**Initial SavingsAccount Balance: 2000.0**

**Deposited: 1000.0**

**Balance after deposit: 3000.0**

**Withdrawal amount exceeds the limit of: 500.0**

**Balance after attempting to withdraw 600: 3000.0**

**Withdrew: 400.0**

**Balance after withdrawing 400: 2600.0**

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.**

**Program code:**

**package Bank.org;**

**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 void setMake(String make) {**

**this.make = make;**

**}**

**public int getYear() {**

**return year;**

**}**

**public void setYear(int year) {**

**this.year = 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 setModel(String model) {**

**this.model = model;**

**}**

**public void displayDetails() {**

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

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

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

**}**

**}**

**public class VehicleD{**

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

**Car car = new Car("Toyota", 2020, "Camry");**

**car.displayDetails();**

**car.setModel("Corolla");**

**car.setYear(2022);**

**System.*out*.println("\nUpdated Car Details:");**

**car.displayDetails();**

**}**

**}**

**Output:**

**Car Make: Toyota**

**Car Year: 2020**

**Car Model: Camry**

**Updated Car Details:**

**Car Make: Toyota**

**Car Year: 2022**

**Car Model: Corolla**

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 code:**

**package Bank.org;**

**class Animal {**

**private String name;**

**public Animal(String name) {**

**this.name = name;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(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(getName() + " is barking.");**

**}**

**}**

**public class AnimalDemo {**

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

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

**animal.eat();**

**animal.sleep();**

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

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

**dog.eat();**

**dog.sleep();**

**dog.bark();**

**dog.setName("Max");**

**System.*out*.println("\nAfter changing the dog's name:");**

**dog.eat();**

**dog.bark();**

**}**

**}**

**Output:**

**Generic Animal is eating.**

**Generic Animal is sleeping.**

**Buddy is eating.**

**Buddy is sleeping.**

**Buddy is barking.**

**After changing the dog's name:**

**Max is eating.**

**Max is barking.**

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

**Program code:**

**package Bank.org;**

**class Student {**

**private String name;**

**private int age;**

**private String grade;**

**public Student(String name, int age, String grade) {**

**this.name = name;**

**this.age = age;**

**this.grade = grade;**

**}**

**public String getName() {**

**return name;**

**}**

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

**this.name = name;**

**}**

**public int getAge() {**

**return age;**

**}**

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

**this.age = age;**

**}**

**public String getGrade() {**

**return grade;**

**}**

**public void setGrade(String grade) {**

**this.grade = grade;**

**}**

**public void displayDetails() {**

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

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

**System.*out*.println("Age: " + getAge());**

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

**}**

**}**

**public class StudentDemo {**

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

**Student student = new Student("Alice", 20, "A");**

**student.displayDetails();**

**student.setName("Bob");**

**student.setAge(21);**

**student.setGrade("B");**

**System.*out*.println("\nUpdated Student Details:");**

**student.displayDetails();**

**}**

**}**

**Output:**

**Student Details:**

**Name: Alice**

**Age: 20**

**Grade: A**

**Updated Student Details:**

**Student Details:**

**Name: Bob**

**Age: 21**

**Grade: B**

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 code:**

**package Demo\_Bank;**

**class Vehicle {**

**private String type;**

**public Vehicle(String type) {**

**this.type = type;**

**}**

**public String getType() {**

**return type;**

**}**

**public void setType(String type) {**

**this.type = type;**

**}**

**public void startEngine() {**

**System.*out*.println("The engine of the " + type + " is starting.");**

**}**

**public void stopEngine() {**

**System.*out*.println("The engine of the " + type + " is stopping.");**

**}**

**}**

**class Car extends Vehicle {**

**public Car(String type) {**

**super(type);**

**}**

**@Override**

**public void startEngine() {**

**System.*out*.println("Car engine is starting with a key ignition.");**

**}**

**@Override**

**public void stopEngine() {**

**System.*out*.println("Car engine is stopping with a key ignition.");**

**}**

**}**

**class Motorcycle extends Vehicle {**

**public Motorcycle(String type) {**

**super(type);**

**}**

**@Override**

**public void startEngine() {**

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

**}**

**@Override**

**public void stopEngine() {**

**System.*out*.println("Motorcycle engine is stopping by turning off the ignition switch.");**

**}**

**}**

**public class Vehical {**

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

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

**car.startEngine();**

**car.stopEngine();**

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

**Motorcycle motorcycle = new Motorcycle("Motorcycle");**

**motorcycle.startEngine();**

**motorcycle.stopEngine();**

**}**

**}**

**Output:**

**Car engine is starting with a key ignition.**

**Car engine is stopping with a key ignition.**

**Motorcycle engine is starting with a kickstart.**

**Motorcycle engine is stopping by turning off the ignition switch.**