CDAC Mumbai PG-DAC August 24

Assignment No- 5

- 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 org.example.demo01;


class Bankaccount {
    private String accountNumber;
    private double balance;


    public Bankaccount(String accountNumber, double
balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }


    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 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: $" +
amount);
        } else {
            System.out.println("Invalid withdrawal
amount.");
        }
    }

    public double getBalance() {
        return balance;
    }

    public String getAccountDetails() {
        return "Account Number: " + accountNumber +
"\nBalance: $" + balance;
    }
}

class SavingsAccount extends Bankaccount {
    private double withdrawalLimit;

    public SavingsAccount(String accountNumber, double
balance, double withdrawalLimit) {
        super(accountNumber, balance);
        this.withdrawalLimit = withdrawalLimit;
    }

    @Override
    public void withdraw(double amount) {
        if (amount > 0 && amount <= withdrawalLimit &&
amount <= getBalance()) {
            super.withdraw(amount);
        } else if (amount > withdrawalLimit) {
            System.out.println("Withdrawal limit
exceeded. Maximum allowed is $" + withdrawalLimit);
        } else {
            System.out.println("Invalid withdrawal
amount.");
        }
    }

    @Override
    public String getAccountDetails() {
        return super.getAccountDetails() + "\nWithdrawal
Limit: $" + withdrawalLimit;
    }
}

public class Main {
    public static void main(String[] args) {
        Bankaccount bankAccount = new
Bankaccount("123456789", 1000.0);
        System.out.println("Bank Account Details:");

    System.out.println(bankAccount.getAccountDetails());
        bankAccount.deposit(200.0);
        bankAccount.withdraw(500.0);
        System.out.println("Balance after transactions:" +
bankAccount.getBalance());
        SavingsAccount savingsAccount = new
SavingsAccount("234554321", 2340.0, 1230.0);
        System.out.println("Savings Account Details:");

    System.out.println(savingsAccount.getAccountDetails());
        savingsAccount.deposit(5450.0);
        savingsAccount.withdraw(23200.0);
        savingsAccount.withdraw(760.0);
        System.out.println("Balance after transactions: " +
savingsAccount.getBalance());
    }
}
```

- 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.

-->>

```
package org.example.demo01;

class Vehicle {
    private String make;
    private int year;

    public Vehicle() {

    }

    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;
    }

    public void displayDetails() {
        System.out.println("Make: " + this.make);
        System.out.println("Year: " + this.year);
    }
}

class Car extends Vehicle {
    private String model;

    public Car() {
    }
    public Car(String make, String model, int year) {
        super(make, year);
        this.model = model;
    }

    public void displayDetail() {
        super.displayDetails();
        System.out.println("Model: " + this.model);
    }
}

public class Demo {
    public static void main(String[] args) {
        Car c = new Car("kia", "muv", 2004);
        c.displayDetail();
    }
}
```

- 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 org.example.demo01;

class Animal {
    private String name;

    public Animal() {
        this("lion");
    }

    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() {
    }

    public Dog(String name) {
        super(name);
    }

    public void bark() {
        System.out.println(getname() + " is barking");
    }
}

public class Program {

    public static void main(String[] args) {
        Dog d = new Dog("Sheru");
        d.eat();
        d.sleep();
        d.bark();
    }
```

```
}
```

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

-->>

```
package org.example.demo01;

class Student {
        private String name;
        private int rollno;
        private int marks;

        public Student() {
                this("Rohan", 5, 97);

        }

        public Student(String name, int rollno, int marks) {
                this.name = name;
                this.rollno = rollno;
                this.marks = marks;
        }

        public String getname() {
                return name;
        }

        public void setname(String name) {
                this.name = name;
        }

        public int getrollno() {
                return rollno;
        }

        public void setrollno() {
                this.rollno = rollno;
```

```
    }

    public int getmarks() {
        return marks;
    }

    public void setmarks() {
        this.marks = marks;
    }

    public void Display() {
        System.out.println("Name is :" + this.name);
        System.out.println("Roll no is : " + this.rollno);
        System.out.println("marks is :" + this.marks);
    }
}

public class Test {
    public static void main(String[] args) {
        Student s = new Student();
        s.Display();
    }
}
```

- 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.

-->>

```
package org.example.demo01;

class Vehicle {
    public void startEngine() {
        System.out.println("Engine is start");
    }

    public void stopEngine() {
        System.out.println("Engine stop");
    }
}

class Car extends Vehicle {
        public void display() {
                super.startEngine();
                super.stopEngine();
        }
}

class Motorcycle extends Vehicle {
        public void display() {
                super.startEngine();
                super.stopEngine();
        }
}

public class Demo1 {
        public static void main(String[] args) {
                Car c = new Car();
                c.startEngine();
                c.stopEngine();
        }
}
```