1. Write a java program to implement the abstraction property.

```
abstract class Animal
{
    abstract void eat();
    abstract void speak();

    void something() {
        System.out.println("Something.");
    }
}

class Dog extends Animal {
    void eat() {
        System.out.println("Dog eats.");
    }

    void speak() {
        System.out.println("Dog barks.");
    }
}

Class P1
{
    public static void main (String args[]) {
        Dog ob = new Dog();
        Ob.eat();
        ob.speak();
        Ob.something();
    }
}
```

OUTPUT

Dog eats.

Dog barks.

Something.

2. Write a java program to implement interface.

```
interface Animal {
        void eat();
        void speak();
}
class Cat implements Animal {
        public void eat(){
                System.out.println("Cat laps milk.");
        }
        public void speak(){
                System.out.println("Cat meows.");
        }
}
Class P2
{
        public static void main(String args[]){
                Cat ob = new Cat();
                ob.eat();
                ob.speak();
        }
}
```

output doesn't match code

OUTPUT

Cat likes milk.

Cat meows.

3. Write a java program to implement multiple inheritance with the help of interface.

→
```
interface AnimalEat {
        void eat();
}
interface AnimalSpeak(){
        void Speak();
}
class Horse implements AnimalSpeak, AnimalEat {
        public void eat() {
                System.out.println("Horse chews straw.");
        }
        public void Speak() {
                System.out.println("Horse neighs.");
        }
}
class P3
{
        public static void main(String args[]) {
                Horse ob = new Horse();
                ob.eat();
                ob.speak();
        }
}
```

OUTPUT

Horse chews straw.
Horse neighs.

4. Write a java program to implement the inheritance in interface.

→
```
interface Animal {
        abstract void eat();
        abstract void speak();
}
Class Rat implements Animal {
        public void eat() {
                System.out.println("Rat eats.");
        }
        public void speak() {
                System.Out.println("Rat squeaks.");
        }
}
class P4 {
        public static void main (String args[]) {
                Rat ob = new Rat();
                ob.eat();
                ob.speak();
        }
}
```

OUTPUT

Rat eats.

Rat Squeaks.

5. Write a java program to implement multiple inheritance using interf

→
```
interface AnimalEat {
        void eat();
}
```

```
interface AnimalSpeak {
        void speak();
}
class Cat implements AnimalSpeak, AnimalEat {
        public void eat() {
                System.out.println("Cat likes milk.");
        }
        public void speak() {
                System.out.println("Cat meows.");
        }
}
class P5 {
        public static void main(String args[]) {
                Cat ob = new Cat();
                Ob.eat();
                ob.speak();
        }
}
```

OUTPUT

Cat likes milk.

Cat meows.

6 Write a java program to implement super keyword in java.

→

```
class K {
        void display() {
                System.out.println("Hello");
        }
}
class L extends K {
        public void meth() {
```

```
        Super. display ();
    }
}
Class P6 {
    public static void main (String args [] ){
        L obj = new L();
        Obj. meth ();
    }
}
```

OUTPUT

Hello

7. Write a java program to implement super() method without parameter.

→

```
class A {
    void display () {
        System.out.println ("Hello ");
    }
}
class B extends A {
    public void meth () {
        Super . display ();
    }
}
class P7 {
    public static void main (String args [] ){
        B obj = new B()
        Obj. meth ();
    }
}
```

OUTPUT

Hello

8 Write a Java program to implement super() method with parameter.

→
```
class A {
    void display (int b){
        System.out.println ("You entered : " + b);
    }
}
class B extends A {
    public void meth (int a) {
        super. display (a);
    }
}
class P8 {
    public static void main (String args[]) {
        B obj = new B();
        obj.meth (7);
    }
}
```

OUTPUT

You entered : 7

9. Implementation of final keyword before a variable.

→
```
class A {
    public void meth() {
        final int K = 10;
        System. out. println (K);
    }
}
```

```
class P9 {
        public static void main (String args[]) {
                new A().meth();
        }
}
```

OUTPUT

10

10. Implementation of final keyword before a method.

→

```
class A {
        final void meth () {
                System.out.println ("Method!");
        }
}
class P10 {
        public static void main (String args[]) {
                new A().meth();
        }
}
```

OUTPUT

Method!

11. Implementation of final keyword before a method.

```
final class A {
        public void meth () {
                System.out.println ("Hello");
        }
}
```

```
class P11 {
        public static void main(String args[]){
                new A(). meth();
        }
}
```

OUTPUT

Hello

12. Create an interface called Player. The interface has an abstract method called play() that display a message describing the meaning of "play" to the class. Create classes called Child, Musician, and Actor that all implement Play. Create an application that demonstrates the use of the classes.

→

```
interface Player {
        void play();
}
class Child implements Player{
        public void play(){
                System.out.println("A Child plays with toys");
        }
}
class Musician implements Player {
        public void play(){
                System.Out.println("A Musician plays an instrument.");
        }
}
class Actor implements Player{
        public void play(){
```

```
        System.out.println("An actor acts in a play.");
    }
}

Class UsePlayer {
    public static void main(String args[]){
        Player ob;
        ob = new Child();
        ob.play();
        ob = new Musician();
        ob.play();
        ob = new Actor();
        ob.play();
    }
}
```

OUTPUT

A child plays with toys.
A Musician plays an instrument.
An actor acts in a play.

13. Create an abstract class Accounts with the following details

Data Members: (a) Balance (b) account Number (c) account Holders Name (d) address

Methods: (a) withdrawl()-abstract, (b) deposit()-abstract
(c) display() to show the balance of the account Number

Create a subclass of this class SavingsAccount and add the following details:

Data Members: (a) rateOf Interest

Methods: (a) calculate Amount()

```
import java.io.*;
abstract class Accounts {
    double balance;
    long accountNumber;
    String accountHolderName = new String();

    Accounts (long ac, String name, double bal, String add) {
        accountNumber = ac;
        balance = bal;
        accountHolders Name = name;
        address = add;
    }

    abstract void withdrawl (double d);
    abstract void deposit (double d);

    void display () {
        System.out.println ("Available Balance: " + balance);
    }
}

class SavingsAccount extends Accounts {
    SavingsAccount (long ac, String name, double bal, String add) {
        super (ac, name, bal, add);
    }

    final double rateOfInterest = 3.5;

    void calculateAmount () {
        super.balance += rateOfInterest /100 * super.balance;
    }
```

address is not defined

```
    void withdrawl (double amt){
            super.balance -= amt;
    }
    void deposit (double amt){
            super.balance += amt;
    }
}

class P13 {
    public static void main (String args[]) throws IOException{
        BufferedReader br = new BufferedReader (new
        InputStreamReader (System.in));

        String n = new String();
        long acn;
        double b, a;
        String add = new String();
        int ch;
        System.out.print("Enter Account Number: ");
        acn = Long.parseLong (br.readLine());

        System.out.print("Enter Name: ");
        n = br.readLine();

        System.out.print("Enter Address: ");
        add = br.readLine();

        System.out.print("Enter Initial Amount: ");
        b = Double.parseDouble (br.readLine());

        SavingsAccount ob = new SavingsAccount (acn, n
                                                b, add);
```

```
while (true) {
    System.out.print ("1. Deposit  2. Withdraw  3. Display
    Balance  4. Exit \n Enter Choice : ");

    ch = Integer. parseInt (br. readLine ());

    switch (ch)
    {
        Case 1 :
        System.out. print (" Enter amount : ");
        a = Double .parse Double (br. readLine());
        ob. deposit (a);
        break ;

        Case 2 :
        System. out. print ("Enter amount : ");
        a = Double . parseDouble (br. readLine());
        ob. withdraw1 (a);
        break ;

        Case 3 :
        Ob. calculateAmount ();
        Ob. display ();
        break ;
        Case 4 :
        System. out. print("Thank You");
        System . exit (0);

        default :
        System. out . println ("Invalid Input");
    }
```

```
        }
    }
}
```

OUTPUT

Enter Account Number : 3045612

Enter Name : Indranil

Enter Address : Kolkata

Enter Initial Amount : 4000

1. Deposit 2. Withdraw 3. Display Balance 4. Exit

Enter Choice : 1

Enter amount : 3000

1. Deposit 2. Withdraw 3. Display Balance 4. Exit

Enter Choice : 2

Enter Amount : 1000

1. Deposit 2. Withdraw 3. Display Balance 4. Exit

Enter Choice : 3

Available Balance : 6210.0

1. Deposit 2. Withdraw 3. Display Balance 4. Exit

Enter Choice : 4

Thank You

11 oct 2022