1.singleton

class Test

{

private static test s=null;

public string str;

private Test()

{

str="Singleton Class";

}

public static test getInstance()

{

if(s ==null)

{

s=null Test();

}

retrun s;

}

}

class Singleton

{

public static void main(String args[])

{

Test a= Test.getInstance();

Test b= Test.getInstance();

a.str=(a.str).toUpperCase();

System.out.println("String-a: "+a.str);

System.out.println("string-b: "+b.str);

System.out.println("Hence, proved.");

}

}

2.organization program

package oop2

public class Employee

{

int id,incentive,overtime;

String name;

double base\_salary;

public Employee(int a,String b,double c){

this.idea;

this.name=b;

this.base\_salary=a;

}

public void salary()

{

double sal=base\_salary;

System.out.println("Base salary is;"+sal);

}

}

public class Manager extends Employee{

double c;

public Manager(int a,String b,double c){

super(a,b,c);

this.c=c;

}

public void salary\_calc(int incentive)

{

double sal=c+insentive;

System.out.println("Manager's Salary is:"+sal);

}

}

public class Labor extends Employee{

double c;

public Labor(int a,String b,double c){

super(a,b,c);

this.c=c;

}

public void salary\_calc(int overtime)

{

double sal=c+overtime;

System.out.println("Labor's Salary is:"+sal);

}

}

public class Organization

{

public static void main(String args[])

{

Manager m = new Manager(123,"Glenn",50000):

m.salary\_calc(5000);

Labor l=new Labor(134,"Abc",10000):

l.salary\_calc(300):

}

}

3.

package oop3

public class Bank {

private String name = "Bank";

int totalAmount;

public void addToTotalBankCash(Bank obj) {

totalAmount += obj.totalAmount;

}

public void showTotal() {

System.out.println(" The total cash in " + name +" is " + totalAmount);

}

public void addAmt(int amt) {

totalAmount += amt;

}

}

public class CurrentAccount extends Bank {

private String name = "Current Account";

public void showtotal() {

System.out.println("The Cash Credits of " + name + " is " + totalAmount);

}

}

public class SavingsAccount extends Bank {

private String name = " Savings Account";

public void showtotal() {

System.out.println("Your Fixed Deposit " + name + " balance is " + totalAmount);

}

}

public class Acc {

public static void main(String[] args) {

Bank newBank = new Bank();

newBank.showTotal();

Bank savingsAc = new SavingsAccount();

Bank current = new CurrentAccount();

savingsAc.addAmt(1000);

current.addAmt(20000);

newBank.addToTotalBankCash(current);

newBank.addToTotalBankCash(savingsAc);

current.showTotal();

savingsAc.showTotal();

newBank.showTotal();

}

}

4.

**package** oop4;

**abstract** **class** Animal {

String name;

**abstract** String bark();

}

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

String bark() {

**return** "BOW BOW";

}

}

**class** Cat **extends** Animal{

String bark() {

**return** "MEOW MEOW";

}

}

**public** **class** Abs {

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

Animal animal=**new** Dog();

// Animal animal=new Cat();

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

}

}

5.SHAPES

package oop5;

public abstract class Draw

{

public abstract void draw();

}

class Line Extends Draw

{

@Override

public void draw()

{

System.out.println("Drawing Line");

}

}

class rectangle Extends Draw

{

@Override

public void draw()

{

System.out.println("Drawing rectangle");

}

}

class cube Extends Draw

{

@Override

public void draw()

{

System.out.println("Drawing cube");

}

}

public class Shapes{

public static void main(String args[]){

Draw d= new line();

d.draw();

Draw d1=new rectangle();

d1.draw();

Draw d2=new cube();

d2.draw();

}

}

6.PERSISTENCE

package persist;

public class Per\_classes

{

}

abstract class persist

{

abstract void per();

}

class filepersistence extends persist

{

@Override

void per()

{

System.out.println("Executing File Persistence");

}

}

class databasepersistence extends persist

{

@Override

void per()

{

System.out.println("Executing database Persistence");

}

}

public class Persistence

{

public static void main(String args[])

{

persist p=new filepersistence();

p.per();

persist p1=new databasepersistence();

p1.per();

}

}

7.