**OPPS ASSIGNMENT**

**1stcode:**

package com.prajwal

class SingleTone {

private static SingleTone stObject;

//private constructor- singleton class will contain a private constructor

private SingleTone() {

}

//while creating object of singleton obj , check it has already an obj, if not create an object.

public static SingleTone getInstance() {

// create object if it's not already created

if(stObject == null) {

stObject = new SingleTone();

}

// returns the singleton object

return stObject;

}

public void getConnection() {

System.out.println("You are now connected to the Singletone Class.");

}

}

class Main {

public static void main(String[] args) {

SingleTone st1;

// refers to the only object of SingleTone

st1= SingleTone.getInstance();

st1.getConnection();

}

}

**2nd code:**

package com.prajwal

public class Employee {

public void managerSalary(double salary, double incentive){

System.out.println("Manager Salary is " +salary + " also include Incentive "+ incentive + "\n");

}

public void labourSalary(double salary, double overtime){

System.out.println("Labour Salary is " +salary + " also include Overtime "+ overtime + "\n");

}

public static void main(String[] args) {

Employee e1 = new Employee();

e1.managerSalary(10000.0, 2000.0);

Manager m1 = new Manager();

m1.managerSalary(10000.0, 2000.0);

Employee e2 = new Employee();

e2.labourSalary(9000.0, 500.0);

Labour l1 = new Labour();

l1.labourSalary(9000.0, 500.0);

}

}

class Manager extends Employee {

@Override

public void managerSalary(double salary, double incentive) {

System.out.println("Manager Salary Sum is " + (salary+incentive) + "\n");

}

}

class Labour extends Employee {

@Override

public void labourSalary(double salary,double overtime) {

System.out.println("Labour Salary Sum is " + (salary + overtime) + "\n");

}

}

**3rd code:**

package com.prajwal

public class BankAccounts {

public double totalAmount,savingsTotal,currentTotal = 0.0;

public static void main(String[] args) {

BankAccounts b1 = new BankAccounts();

BankAccounts s1 = new SavingsAccount();

BankAccounts c1 = new CurrentAccount();

s1.deposit("savings",1000.0);

s1.deposit("savings",1000.0);

c1.deposit("current",1000.0);

c1.deposit("current",1000.0);

c1.deposit("current",1000.0);

b1.getTotalBankAmount(s1.savingsTotal,c1.currentTotal);

}

public void deposit(String accountType, double amount){

// System.out.println("Account Type: " +accountType);

// currentTotal =+ amount;

}

public void getTotalBankAmount(double s,double c){

System.out.println("\nTotal Savings Amount: " + s +" & Total Current Amount : "+ c);

System.out.println("Total bank Amount is " + (s+c));

}

}

class SavingsAccount extends BankAccounts{

//method to deposit to Savings Account

@Override

public void deposit(String accountType, double amount) {

savingsTotal = savingsTotal + amount;

System.out.println("Account Type: " +accountType + ", Savings Total :" +savingsTotal + "\n");

}

}

class CurrentAccount extends BankAccounts{

//method to deposit to Current Account

@Override

public void deposit(String accountType, double amount){

currentTotal = currentTotal+ amount;

System.out.println("Account Type: " +accountType + ", Current Total :" +currentTotal + "\n");

}

}

**4th code:**

package com.prajwal

public class AbstractClass {

public static void main(String[] args) {

Pig myPig = new Pig(); // Create a Pig object

myPig.animalSound();

myPig.sleep();

}

}

// Abstract class

abstract class Animal {

// Abstract method (does not have a body)

public abstract void animalSound();

// Regular method

public void sleep() {

System.out.println("Zzz");

}

}

//abstract class cant inherit so create a extended class and define abstract method here

// Subclass (inherit from Animal)

class Pig extends Animal {

public void animalSound() {

// The body of animalSound() is provided here

System.out.println("The pig says: wee wee");

}

}

**5th code:**

package com.prajwal

abstract class Shape {

public abstract void draw();

public static void main(String[] args) {

Rectangle R1 =new Rectangle();

R1.draw();

Cube C1 = new Cube();

C1.draw();

Line L1 = new Line();

L1.draw();

}

}

class Rectangle extends Shape{

@Override

public void draw() {

System.out.println("Draw Rectangle");

}

}

class Line extends Shape{

@Override

public void draw() {

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

}

}

class Cube extends Shape{

@Override

public void draw() {

System.out.println("Draw Cube");

}

}

**6th code:**

package com.prajwal

abstract class Persistence {

public abstract void persist();

public static void main(String[] args) {

Persistence p1 = new FilePersistence();

p1.persist();

}

}

class FilePersistence extends Persistence {

@Override

public void persist() {

System.out.println("inside FilePersistence");

}

}

class DatabasePersistence extends Persistence {

@Override

public void persist() {

System.out.println("inside DatabasePersistence");

}

}