OOPS ASSIGNMENT

2. class Employee{

float salary = 40000;

void totalSalary()

{

System.out.println("The Employee incremented salary is :" +(salary ) );

}

}

class Manager extends Employee{

float incentive =15000;

void totalsalary()

{

System.out.println("The Manager salary is :" +((salary+incentive)));

}

}

class Labour extends Manager{

double overtime=6500;

void totalsalary()

{

System.out.println("The salary of the Labours is"+(overtime+salary));

}

}

public class MainClass90

{

public static void main(String args[]){

Labour t = new Labour();

Manager m=new Manager();

System.out.println("Total Salary Of Employees In Organisation is "+(m.salary+ (((m.incentive)+ t.overtime))+ t.salary)); //displays total salary of emp in org//

// based on an object it decides which class salary method to be execute

// m.totalsalary(); // displays manager salary//

// t.totalsalary(); //displays labour salary//

}

}

3. class Bank{

void totalMoneyInBank()

{

System.out.println("The Employee incremented salary is :" );

}

}

class SavingAcoount extends Bank{

float fixedDeposits =150000;

void totalMoneyInBank()

{

System.out.println("total money in savings account :" +((fixedDeposits)));

}

}

class CurrentAcccount extends SavingAcoount{

double cashCredit=59000;

void totalMoneyInBank()

{

System.out.println("The total money in current account "+(cashCredit));

}

}

public class MainClass90

{

public static void main(String args[]){

CurrentAcccount m=new CurrentAcccount();

SavingAcoount s=new SavingAcoount();

// based on an object it decides which class account method to be execute

m.totalMoneyInBank(); //prints total money in currect ac//

s.totalMoneyInBank(); //prints total money in savings ac//

System.out.println("Total money in the bank is "+( m.fixedDeposits+m.cashCredit)); //displays total salary of emp in org//

}

}

6.. package oops;

abstract class Persistence{

abstract void persist();

}

class FilePersistence extends Persistence{

public void persist() {

System.out.print("Persist is in FilePersistence");

}

}

class DatabasePersistence extends Persistence{

public void persist() {

System.out.print("Persist is in DatabasePersistence");

}

}

public class Abstract {

public static void main(String args[]) {

FilePersistence F=new FilePersistence();

F.persist();

DatabasePersistence D=new DatabasePersistence();

D.persist();

}

}

7 public class Cookie extends DessertItem

{

private int quantity;

private double pricePerDozen;

public Cookie()

{

super();

quantity = 0;

pricePerDozen = 0;

}

public Cookie(String name, int qty, double pri)

{

super(name);

quantity = qty;

pricePerDozen = pri;

}

public int getQuantity() {

return quantity;

}

public double getPricePerDozen() {

return pricePerDozen;

}

public void setPricePerDozen(double pricePerDozen) {

this.pricePerDozen = pricePerDozen;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

// @Override

public double getCost() {

double total = pricePerDozen / 12 \* quantity;

total = Math.round(total \* 100);

return total;

}

public String toString()

{

String s = String.format("%-50s $%.2f\n\t %d cookies @ $%.2f per Dozen", getName(), getCost()/100, quantity, pricePerDozen);

return s;

}

}

7.

public class Candy extends DessertItem

{

private double weight;

private double pricePerPound;

public Candy()

{

super();

weight = 0;

pricePerPound = 0;

}

public Candy(String name, double w, double pri)

{

super(name);

weight = w;

pricePerPound = pri;

}

public double getWeight() {

return weight;

}

public void setWeight(double weight) {

this.weight = weight;

}

public double getPricePerPound() {

return pricePerPound;

}

public void setPricePerPound(double pricePerPound) {

this.pricePerPound = pricePerPound;

}

// @Override

public double getCost() {

double total = weight \* pricePerPound;

total = Math.round(total \* 100);

return total;

}

public String toString()

{

String s = String.format("%-50s $%.2f\n\t %.2f lbs @ $.2f", getName(), getCost()/100, weight, pricePerPound);

return s;

}

}