**OOPS ASSIGNMENT**

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();

}

}

**CHOCOLATE EXAMPLE IN OOPS ASSIGNMENT**

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;

}

}

abstract class DessertItem

{

protected String name;

public DessertItem()

{

name = "";

}

public DessertItem(String name1)

{

name = name1;

}

public String getName()

{

return name;

}

public void setName(String name1)

{

name = name1;

}

public abstract double getCost();

}

public class DessertShop

{

public static void main(String[] args)

{

Candy item1 = new Candy("Peanut Butter Fudge", 2.25, 3.99);

Cookie item2 = new Cookie("Oatmeal Raisin Cookies", 4, 3.99);

IceCream item3 = new IceCream("Vanilla Ice Cream", 2, 1.05, 0.45);

System.out.println(item1);S

System.out.println(item2);

System.out.println(item3);

}

}

public class IceCream extends DessertItem

{

private int numberOfScoops;

private double pricePerScoop;

private double toppingPrice;

public IceCream()

{

super();

numberOfScoops = 0;

pricePerScoop = 0;

toppingPrice = 0;

}

public IceCream(String name, int scoops, double prcPerScoop, double toppings)

{

super(name);

numberOfScoops = scoops;

pricePerScoop = prcPerScoop;

toppingPrice = toppings;

}

public int getNumberOfScoops() {

return numberOfScoops;

}

public void setNumberOfScoops(int numberOfScoops) {

this.numberOfScoops = numberOfScoops;

}

public double getPricePerScoop() {

return pricePerScoop;

}

public void setPricePerScoop(double pricePerScoop) {

this.pricePerScoop = pricePerScoop;

}

public double getToppingPrice() {

return toppingPrice;

}

public void setToppingPrice(double toppingPrice) {

this.toppingPrice = toppingPrice;

}

@Override

public double getCost() {

double total = (numberOfScoops \* pricePerScoop + toppingPrice);

return Math.round(100 \* total );

}

public String toString()

{

String s = String.format("%-50s $%.2f\n\t %d scoops @ $%.2f/scoop + $%.2f", getName(), getCost()/100, numberOfScoops, pricePerScoop, toppingPrice);

return s;

}

}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;

}

}