

OOP Lab 9

Topic: Inheritance & Polymorphism

Marks: 30 (10 marks for each class's correct implementation)

Release Date: 25-Dec-20 Friday

Submission Date & Time: 26-Dec-20, 11:59 PM Saturday

You may send your lab queries at: asktoknow42@gmail.com

Task 1 (30 pts)

Identify classes and attributes and Implement the following problem.

There is a Dessert Shoppe which sells cookies by the dozen, ice cream, and sundaes (ice cream with a topping). Your code will be used for the checkout system.

To do this, you will implement an inheritance hierarchy of classes derived from a **DessertItem** superclass.

The **Cookie**, and **IceCream** classes will be derived from the **DessertItem** class.

The **Sundae** class will be derived from the **IceCream** class.

The DessertItem Class

The **DessertItem** class is a *superclass* from which specific types of **DessertItems** can be derived. It contains only one data member, a name (char *). It also defines a number of methods. All of the **DessertItem** class methods except the `getCost()` method are defined in a generic way.

The `getCost()` method in **DessertItem** class should only return 0 because the method of determining the costs varies based on the type of item.

Tax amounts should be rounded to the nearest Paisa. For example, calculating the tax on a food item with a cost of 199 paisa with a tax rate of 2.0% should be 4 paisa.

The Derived Classes

All of the classes which are derived from the **DessertItem** class must define a constructor.

The **Cookie** class should be derived from the **DessertItem** class. A **Cookie** item has a *number* and a *price per dozen* which are used to determine its *cost*. For example, 4 cookies @ 399 paisa /dz. = 133 paisa. The cost should be rounded to the nearest paisa.

The **IceCream** class should be derived from the **DessertItem** class. An **IceCream** item simply has a *cost*.

The **Sundae** class should be derived from the **IceCream** class. The *cost* of a Sundae is the *cost of the IceCream* plus the *cost of the topping*.

Make the two functions `getCost()` `getTax()` virtual in base class.

Add attributes/functions where needed to implement this system.

In main user should be able to add items to a customer list. You need to keep track of list of items and your program should return total cost of a bill including tax.

```
int main()
{
    int itemCount = 0;
    DessertItem* itemList[20];    //20 pointers of base class type, at most 20 objects
                                   //can be added

    char addMore;
    char userChoice;

    cout<<"Want to add more items\n";
    cin>> addMore;    //enter y for yes
    while(addMore == 'y')
    {
        cout<<"Choose Item you want to add\n";
        if(userChoice==1)
        {
            //take parameters from user needed for a Cookie
            ...
            itemList[itemCount++]= new Cookie(...)    //making base class pointer
                                                         //point cookie object
        }
        else if(userChoice==2)
        {
            //take parameters from user needed for an IceCream
            ...
            itemList[itemCount++]= new IceCream(...)    //making base class pointer
                                                         //point IceCream object
        }
        else if(userChoice==3)
        {
            //take parameters from user needed for a Sundae
            ...
            itemList[itemCount++]= new Sundae(...)    //making base class pointer
                                                         //point Sundae object
        }
        cout<<"Want to add more items\n";
        cin>> addMore;
    }

    int TotalCost;
    //Find totalCost (sum of costs of all items in list without tax) and print it.
    ...

    int TotalTax;
    //Find totalCost (sum of taxes of all items in list) and print it.
    ...
}
```