

```

import java.io.*;
import java.text.DecimalFormat;

//////////////////////////////////////////////////////////////////
//
// The following program uses Inheritance and overriding methods to model //
// a grocery store. //
// //
// WRITTEN BY: //
// ALAIN DADAIAN //
// //
//////////////////////////////////////////////////////////////////

class Hw14
{
    static Item [] items;

    //-----
    // The constructor Hw14() contains the list of items and puts them
    // into the array items.
    //-----
    public Hw14()
    {
        items = new Item[7];

        items[0] = new UnitItem("tissue", 1.22);
        items[1] = new BulkItem("chicken,wings", 2.57);
        items[2] = new BulkItem("chicken,whole", 1.80);
        items[3] = new UnitItem("jello", 2.89);
        items[4] = new UnitItem("crackers", 2.95);
        items[5] = new BulkItem("bananas", 1.42);
        items[6] = new UnitItem("broth", 0.97);
    }

    //-----
    public static void main(String [] args) throws Exception
    {
        Hw14 list = new Hw14();

        String name = "";
        double price, pounds, cost, total = 0.0;
        int quantity, counter = 0;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        DecimalFormat fmt = new DecimalFormat ("####0.00");

        System.out.println();

        while (counter < 100)
        {
            System.out.print("Item? ");
            name = br.readLine();

            if (name.equals("quit"))
                break;

            while (counter <= items.length)
            {
                if (counter == 7)
                    break;
                else
                {
                    if (!(name.equals(items[counter].getName())))
                        counter++;
                    else
                        break;
                }
            }

            if (counter == items.length && !name.equals("quit"))
            {
                System.out.println("Sorry, but we don't carry that item here.\n");
                counter = 0;
            }
            else
            {
                cost = items[counter].askMultiplyPrint();
                System.out.println("Cost is " + fmt.format(cost) + "\n");
            }
        }
    }
}

```

```

        total += cost;
        counter = 0;
    }
}

System.out.println("\nYour total cost is " + fmt.format(total));
System.out.println("Thank you for shopping with Java!\n");

}
//-----

} // end of class Hw14

////////////////////////////////////

class UnitItem extends Item
{
    protected double price;

    //-----
    // The UnitItem constructor has two arguments a String name and
    // a double price. It uses the super reference to get the String
    // name to the Item class.
    //-----
    public UnitItem(String name, double price)
    {
        super (name);

        this.price = price;
    }

    //-----
    // The askMultiplyPrint method is an overriding method. In this
    // case because it is used under the UnitItem class it asks for
    // the quantity the user desires which is read as an integer and
    // returns the quantity.
    //-----
    public double askMultiplyPrint() throws Exception
    {
        int quantity;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Quantity desired? ");
        quantity = Integer.parseInt(br.readLine());

        return price * quantity;
    }
    //-----

} // end of class UnitItem

////////////////////////////////////

class BulkItem extends Item
{
    protected double price;

    //-----
    // The BulkItem constructor has two arguments a String name and
    // a double price. It uses the super reference to get the String
    // name to the Item class.
    //-----
    public BulkItem(String name, double price)
    {
        super (name);

        this.price = price;
    }

    //-----
    // The askMultiplyPrint method is an overriding method. In this
    // case because it is used under the BulkItem class it asks for
    // the number of pounds the user desires which is read as a double
    // and returns the number of pounds.
    //-----
    public double askMultiplyPrint() throws Exception
    {
        double pounds;

```

```

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Pounds desired? ");
        pounds = Double.parseDouble(br.readLine());

        return price * pounds;
    }
    //-----
} // end of class BulkItem

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

abstract class Item
{
    protected String name;

    //-----
    // The item constructor takes one argument a name of type String.
    //-----
    public Item(String name)
    {
        this.name = name;
    }

    //-----
    // Returns the name of the item desired.
    //-----
    public String getName()
    {
        return name;
    }

    //-----
    // The UnitItem and BulkItem classes must define the
    // askMultiplyPrint method for there certain types of products.
    //-----
    public abstract double askMultiplyPrint() throws Exception;

    //-----
} // end of class Item

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```