

A Shopping Cart

In this exercise you will complete a class that implements a shopping cart as an array of items. The file *Item.java* contains the definition of a class named *Item* that models an item one would purchase. An item has a name, price, and quantity (the quantity purchased). The file *ShoppingCart.java* implements the shopping cart as an array of *Item* objects.

1. Complete the *ShoppingCart* class by doing the following:
 - a. Declare an instance variable *cart* to be an array of *Items* and instantiate *cart* in the constructor to be an array holding *capacity* *Items*.
 - b. Fill in the code for the *increaseSize* method. Your code should be similar to that in Listing 7.8 of the text but instead of doubling the size just increase it by 3 elements.
 - c. Fill in the code for the *addToCart* method. This method should add the item to the cart and update the *totalPrice* instance variable (note this variable takes into account the quantity).
 - d. Compile your class.
2. Write a program that simulates shopping. The program should have a loop that continues as long as the user wants to shop. Each time through the loop read in the name, price, and quantity of the item the user wants to add to the cart. After adding an item to the cart, the cart contents should be printed. After the loop print a "Please pay ..." message with the total price of the items in the cart.

```
// *****

//   Item.java

//

//   Represents an item in a shopping cart.

// *****

import java.text.NumberFormat;

public class Item

{

    private String name;

    private double price;

    private int quantity;
```

```

// -----
//  Create a new item with the given attributes.
// -----
public Item (String itemName, double itemPrice, int numPurchased)
{
    name = itemName;
    price = itemPrice;
    quantity = numPurchased;
}

// -----
//  Return a string with the information about the item
// -----
public String toString ()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    return (name + "\t" + fmt.format(price) + "\t" + quantity + "\t"
            + fmt.format(price*quantity));
}

// -----
//  Returns the unit price of the item
// -----
public double getPrice()
{
    return price;
}

// -----
//  Returns the name of the item

```

```

// -----
public String getName()
{
    return name;
}

// -----
// Returns the quantity of the item
// -----
public int getQuantity()
{
    return quantity;
}
}

// *****
// ShoppingCart.java
//
// Represents a shopping cart as an array of items
// *****

import java.text.NumberFormat;

public class ShoppingCart
{
    private int itemCount;        // total number of items in the cart
    private double totalPrice;    // total price of items in the cart

```

```

private int capacity;          // current cart capacity

// -----
//  Creates an empty shopping cart with a capacity of 5 items.
// -----
public ShoppingCart()
{
    capacity = 5;
    itemCount = 0;
    totalPrice = 0.0;
}

// -----
//  Adds an item to the shopping cart.
// -----
public void addToCart(String itemName, double price, int quantity)
{
}

// -----
//  Returns the contents of the cart together with
//  summary information.
// -----
public String toString()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    String contents = "\nShopping Cart\n";
    contents += "\nItem\t\tUnit Price\tQuantity\tTotal\n";

    for (int i = 0; i < itemCount; i++)

```

```
        contents += cart[i].toString() + "\n";

        contents += "\nTotal Price: " + fmt.format(totalPrice);
        contents += "\n";

        return contents;
    }

    // -----
    //  Increases the capacity of the shopping cart by 3
    // -----
    private void increaseSize()
    {
    }
}
```