

## Invoice Class

- 1) Create a class called *Invoice* that a hardware store might use to represent an invoice for an item sold at the store.
- 2) An *Invoice* should include four pieces of information—
  - a) a part number (type *string*) as a public auto-implemented property,
  - b) a part description (type *string*) as public auto-implemented property,
  - c) a quantity of the item being purchased (type *int*) as a private field, and
  - d) a price per item (*decimal*) as a private field.
- e) Use *PascalCase* to name a *property* and *\_camelCase* to name a *private field*.
- 3) In the *Invoice* class Provide a property with a *get* and *set* accessor for the *Quantity* and *PricePerItem* properties by using their backing fields. Also, if the value passed to the *set* accessor is negative, the value of the instance variable should be left unchanged.
- 4) The *Invoice* class should have a constructor that initializes the four Properties in the same order as 2a—2d from the arguments to the constructor.
- 5) In the *Invoice* class provide a method named *GetInvoiceAmount* that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a *decimal* value.
- 6) In the *InvoiceTest* class's *Main* method, write some codes that demonstrate class *Invoice*'s capabilities below:
  - a) Prompt to input the *four pieces of information* in an *Invoice*.
  - b) Display the *four pieces of information* from the *Invoice* object. You will print the following information on four separate lines:
    - i) Part number,
    - ii) Part description,
    - iii) Quantity, and
    - iv) Price per item.
  - c) Prompt the user to enter *four other pieces of information*.
  - d) Update the invoice object with those *four other pieces of information*.
  - e) Display the *four pieces of information* from the updated *invoice* object.
- 7) Make sure all your unit tests for *Invoice* passes.
  - a) *TestInvoiceClass* that tests the structure of the *Invoice* class;
  - b) *TestInvoiceObject* that tests the behaviors and states of the *Invoice* instances; and
  - c) *TestInvoiceProgram* that tests the program by feeding test cases and asserting against known outputs.

```
mikeywu — Visual Studio External Console —...
Enter the part number, part description, quantity, and price per item on separate lines:
001234
Hammer
451
1.79
Original invoice information
Part number: 001234
Description: Hammer
Quantity: 451
Price: $1.79
Invoice amount: $807.29
Update the part number:
ABCDEF
Update the part description
Golf Ball
Update the quantity
-1
Update the price per item
3.21

Updated invoice information
Part number: ABCDEF
Description: Golf Ball
Quantity: 451
Price: $3.21
Invoice amount: $1,447.71
```