



# ITDUMPS

## QUESTION & ANSWER

Accurate study guides, High passing rate!  
IT dumps provides update free of charge  
in one year!

[HTTP://WWW.ITDUMPS.COM](http://www.itdumps.com)

**Exam** : **070-484**

**Title** : Essentials of Developing  
Windows Store Apps using  
C#

**Version** : DEMO



Database that stores information about a purchase

User interaction that is tracked by capturing gestures

Answer Area

Layer	Component
Presentation	XAML files for the user interface
Business	Workflow rules that establish how a purchase transaction notifies the distribution warehouse for shipping
Data	Components that retrieve product information from the data store

2.DRAG DROP

You are developing a Windows Store app.

The app will use a model that is defined by using the following code:

```
public class Customer
{
    private string fName;
    private string lName;
    public Customer(string first, string last)
    {
        this.fName = first;
        this.lName = last;
    }
    public string FirstName
    {
        get { return fName; }
        set { fName = value; }
    }
    public string LastName
    {
        get { return lName; }
        set { lName = value; }
    }
}
```

You need to create a class to represent a collection of Customer objects. The class will be used for data binding. The solution must ensure that if changes are made to the objects of the class, a notification will be sent to the user interface (UI) controls to which the collection is bound.

You have the following signature:

```
public class CustomerList: Target 1<Target 2>
```

Which elements should you include in Target 1 and Target 2 to complete the signature for the collection class? (To answer, drag the appropriate elements to the correct targets. Each element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Elements	Answer Area
Customer	Target 1: Element
CustomerCollection	Target 2: Element
HashTable	
INotifyPropertyChanged	
List	
ObservableCollection	

**Answer:**

The screenshot shows an exam interface. On the left, under the heading 'Elements', there is a list of four items: 'CustomerCollection', 'HashTable', 'INotifyPropertyChanged', and 'List'. On the right, under the heading 'Answer Area', there are two target boxes. 'Target 1:' contains the text 'ObservableCollection'. 'Target 2:' contains the text 'Customer'.

**Explanation:**

\* In C# and Visual Basic, the generic `ObservableCollection<T>` class is a good collection choice for data binding, because it implements the `INotifyPropertyChanged` and `INotifyCollectionChanged` interfaces.

**3 HOTSPOT**

A class named `AccountViewModel` includes a property named `Name` that will be bound to a control.

The `Name` property will occasionally be updated programmatically. The updated values must be reflected in the bound control. You need to implement the interface so that the `AccountViewModel` class can inform WinRT when there is a new value to display.

You have the following code:

```
class AccountViewModel Target 1
{
    public string Name
    {
        get ...
        set ...
    }
    protected void OnPropertyChanged (string name)...
    public event Target 2
    private string _name;
}
```

Which code snippets should you insert in Target 1 and Target 2 to complete the code? (To answer, select the correct code snippet from each drop-down list in the answer area.)

The screenshot shows the 'Answer Area' with two drop-down lists. The first is labeled 'Target 1:' and the second is labeled 'Target 2:'. Both lists are currently empty.

**Answer Area**

Target 1:

- : IBindablePropertyChanged
- : INotifyPropertyChanged
- : INotifyPropertyChanging
- : IBindablePropertyChanging

Target 2:

- BindablePropertyChangedEventHandler PropertyChanged;
- PropertyChangedEventHandler PropertyChanged;
- NotifyPropertyChangingEventHandler PropertyChanged;
- BindablePropertyChangingEventHandler PropertyChanged;

**Answer:**

**Answer Area**

Target 1:

- : IBindablePropertyChanged
- : INotifyPropertyChanged**
- : INotifyPropertyChanging
- : IBindablePropertyChanging

Target 2:

- BindablePropertyChangedEventHandler PropertyChanged;
- PropertyChangedEventHandler PropertyChanged;**
- NotifyPropertyChangingEventHandler PropertyChanged;
- BindablePropertyChangingEventHandler PropertyChanged;

**Explanation:**

\* INotifyPropertyChanged

The INotifyPropertyChanged interface is used to notify clients, typically binding clients, that a property value has changed.

\* PropertyChangedEventHandler

Example:

```
// This is a simple customer class that
// implements the IPropertyChange interface.
public class DemoCustomer : INotifyPropertyChanged
{
    // These fields hold the values for the public properties.
    private Guid idValue = Guid.NewGuid();
    private string customerNameValue = String.Empty;
    private string phoneNumberValue = String.Empty;

    public event PropertyChangedEventHandler PropertyChanged;

    // This method is called by the Set accessor of each property.
    // The CallerMemberName attribute that is applied to the optional propertyName
    // parameter causes the property name of the caller to be substituted as an argument.
```

4.You are developing a Windows Store app. You need to create a certificate to sign the app in a test environment.

Which tool or tools should you use?

- A. The Authorization Manager snap-in
- B. The Certification Authority console
- C. Microsoft Visual Studio 2013
- D. The Security Templates snap-in

**Answer: B**

5.You are developing a Windows Store app. You need to ensure that the app meets the requirements for Windows Store certification.

Which requirement must be met?

- A. The app must encrypt all personally identifiable information.
- B. The app must have a short name and a long name.
- C. The trial functionality must resemble the actual functionality of the app.
- D. The app must conform to the rating system E, Y-7, Y-14, MA.

**Answer: C**