




 My Home
 > ITD-2263-0 - Graphica...
  |  | 
 Sharyl Hammer



My Home Calendar FAQ Help My Library Logged in as sharylhw
 Log Out 12/9/2016

ITD-2263-0 - GRAPHICAL USER INTERFACE DEV OKM.60456.201660

Course Home Content Discussions Dropbox Quizzes Classlist Edit Course
 Grades Blackboard Collaborate

Quiz Submissions - Chapter 13 Quiz ▾

Sharyl Hammer (username: sharylhw)

Attempt 1

Written: Oct 6, 2016 10:59 AM - Oct 6, 2016 12:06 PM

Submission View

Your quiz has been submitted successfully.

Chapter 13 Random

Question 1

1 / 1 point

When you code a statement in a form class that uses a property that throws an argument exception if the argument that's passed to it is invalid, you should

- ☐ use a try-catch statement to catch the exception that's thrown
- ☐ rely on the property's data validation
- ☒ validate the argument before it is passed to the property so the exception is never thrown
- ☐ use a delegate to refer to the exception handler for the argument that's thrown

Question 2

0 / 1 point

Code example 13-1

```
public class CustomerList
{
    private List<Customer> customers;

    public delegate void ChangeHandler(CustomerList customers);
    public event ChangeHandler ChangedList;
```

```

public CustomerList()
{
    customers = new List<Customer>();
}

public void Add(Customer c)
{
    customers.Add(c);
}

public static CustomerList operator + (CustomerList customers, Customer c)
{
    customers.Add(c);
    return customers;
}
}

```

(Refer to code example 13-1) Write the declaration for an indexer for the CustomerList class that uses an int value named i to get or set a Customer object at the specified index.

Answer: `public Customer this[int i] { get { return customers[i]; } set { customers[i] = value; ; } }`

 `//public Customer this [int i]/`

Question 3

0 / 1 point

Code example 13-1

```

public class CustomerList
{
    private List<Customer> customers;

    public delegate void ChangeHandler(CustomerList customers);
    public event ChangeHandler ChangedList;

    public CustomerList()
    {
        customers = new List<Customer>();
    }

    public void Add(Customer c)
    {
        customers.Add(c);
    }
}

```


```

    public static CustomerList operator + (CustomerList customers, Customer c)
    {
        customers.Add(c);
        return customers;
    }
}

```

(Refer to code example 13-1.) Suppose that you've used the Load event handler for a form to add the wiring for the event in the CustomerList class so it should be handled by an event handler named CustomerListChange. Write the declaration for this event handler assuming that you've used customers as the name for the CustomerList object.

Answer: `customers.Changed += new CustomerListChange.ChangeHandler(HandleChange);`

 `(/private void CustomerListChange(CustomerList customers)/)`

Question 4

1 / 1 point

Code example 13-1

```

public class CustomerList
{
    private List<Customer> customers;

    public delegate void ChangeHandler(CustomerList customers);
    public event ChangeHandler ChangedList;


    public CustomerList()
    {
        customers = new List<Customer>();
    }

    public void Add(Customer c)
    {
        customers.Add(c);
    }

    public static CustomerList operator + (CustomerList customers, Customer c)
    {
        customers.Add(c);
        return customers;
    }
}

```

(Refer to code example 13-1) Write the code for activating the event that's declared for the CustomerList class. Assume that you're going to add this code to the Add method.

Answer: ChangedList(this);  (/ChangedList(this);/)


Question 5

1 / 1 point

Code example 13-2

```
customer.NameChanged += new EventHandler (Customer_NameChanged) ;
```


(Refer to code example 13-2.) What is the name of the delegate?

- ☐ Customer
- ☐ NameChanged
- ☐ Customer_NameChanged
-  ☒ EventHandler

Question 6

1 / 1 point


If an invalid argument is passed to the set accessor of a property, the property should throw a/an _____.

Answer: argument exception 

Question 7

1 / 1 point


If you overload the == operator, you must also overload the _____ operator.

Answer: != 

Question 8

1 / 1 point

An indexer

- ☐ is a special type of property
- ☐ uses the *this* keyword in its declaration
- ☐ lets the user of a class access an item by using an index
-  ☒ all of the above


Question 9

1 / 1 point

Code example 13-2

```
customer.NameChanged += new EventHandler (Customer_NameChanged) ;
```

(Refer to code example 13-2.) What is the name of the event?

- ☐ Customer
-  ☒ NameChanged
- ☐ Customer_NameChanged
- ☐ EventHandler

Question 10

0 / 1 point

Code example 13-1

```
public class CustomerList
{
    private List<Customer> customers;

    public delegate void ChangeHandler(CustomerList customers);
    public event ChangeHandler ChangedList;

    public CustomerList()
    {
        customers = new List<Customer>();
    }

    public void Add(Customer c)
    {
        customers.Add(c);
    }

    public static CustomerList operator + (CustomerList customers, Customer c)
    {
        customers.Add(c);
        return customers;
    }
}
```

(Refer to code example 13-1.) As you can see, the CustomerList class overloads the binary + operator to make it easier for you to add Customer objects to a CustomerList object. Now, write code that uses the += operator to add a Customer object named newCustomer to a CustomerList object named customers.

Answer: customers += c;  (/customers += newCustomer;/)

Attempt Score: 7 / 10

Overall Grade (highest attempt): 7 / 10

Done

