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

My Home Calendar FAQ Help My Library Logged in as sharylw

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 - Exam 1 ▾

Sharyl Hammer (username: sharylw)

Attempt 1

Written: Oct 6, 2016 2:30 PM - Oct 6, 2016 4:44 PM

Submission View

Your quiz has been submitted successfully.

Chapter 12 Random

Question 1 **1 / 1 point**

To review the members of two or more classes at the same time, you can use

○ the Solution Explorer
○ the Class Details window
● a class diagram
○ the Class Designer toolbar

Question 2 **1 / 1 point**

When a programmer hides some of the data and operations of a class while she exposes others, she’s using the concept of _______________.

Answer: encapsulation

Question 3 **1 / 1 point**

What happens when this statement is executed?

```
Automobile car = new Automobile(1);
```

● An object of the Automobile class is created.

○ An object of the Automobile class is declared, but not created.

○ A constructor that accepts a string in the Automobile class is called.

○ The default constructor in the Automobile class is called.

Question 4 **0 / 1 point**

Code a statement that will get the value of a public static field named Count that's defined in the Account class, and store the value in a new int variable named count. Assume that you've already created an object from this class that's named account.

____

Answer: public static int count = account.Count;

Question 5 **1 / 1 point**

When you're entering the C# code for a form and want to refer to a static method in another class, you start with

● the name of the class that contains it

○ the name of an object created from the class that contains it

○ the name of the method

Question 6 **1 / 1 point**

The Generate From Usage feature lets you generate a code stub for a class or member from

○ the Solution Explorer

○ the Exception Assistant

○ the Generate From Usage wizard

● the smart tag menu

Question 7 **1 / 1 point**

Which of the following is *not* true about a structure?

○ A structure requires less memory and instantiates faster than a class.

○ You can code members for a structure just as you can for a class.

○ You can code more than one constructor for a structure.

● A structure lets you instantiate a reference type just like a class.

Question 8 **1 / 1 point**

To generate the starting code for a new member of a class, you can use

○ the Solution Explorer

● the Class Details window

○ a class diagram

○ the Class Designer toolbar

Question 9 **0 / 1 point**

Static methods can be called without creating an instance of a ____________.

Answer: class

Question 10 **0 / 1 point**

Code a statement that creates an instance of an Account class using the default constructor and stores the object that’s created in a variable named account.

____

Answer: public Account { };

Question 11 **1 / 1 point**

A class defines the properties and methods of

○ a structure

○ a value

● an object

○ a member

Question 12 **1 / 1 point**

What feature are you taking advantage of when you call the ToDecimal method of the Convert class without knowing how it’s coded?

○ instantiation

● encapsulation

○ inheritance

○ modeling

Question 13 **1 / 1 point**

Which method is an example of overloading the method that follows?

```
public int ParseNumber(String numberString){...}
```

○ `public int Parse(String numberString){...}`

○ `public int ParseNumber(String num){...}`

● `public int ParseNumber(String numberString, String entry){...}`

Question 14 **0 / 1 point**

The data of a class is stored in the class's _____________.

Answer: object

Question 15 **0 / 1 point**

Within a class, you can code a _________________ to create an instance of the class and initialize its instance variables.

Answer: constructor

Question 16 **1 / 1 point**

To create an instance of a class and assign values to it in a single statement without explicitly calling a constructor, you can use a/an _____________________.

Answer: object initializer

Question 17 **1 / 1 point**

The process of creating an object from a class is known as

● instantiation

○ encapsulation

○ inheritance

○ modeling

Question 18 **0 / 1 point**

Code a statement that sets the value of the Age property of an Account object named account to the value in a variable named newAge.

____

Answer: public int newAge = account.Age;

Question 19 **1 / 1 point**

To begin the declaration for a constructor, you code the public keyword followed by

○ the name of the property

○ the data type of the class

● the name of the class

○ the arguments

Question 20 **1 / 1 point**

To make a method in a class available to other classes, you begin the method declaration with

● the public keyword

○ the private keyword

○ the void keyword

○ the return keyword

Question 21 **1 / 1 point**

When you design and develop business classes for an application, your goal is to

○ allow development to be spread among members of a development team

○ separate the business rules from the presentation and database logic

○ make the application easier to develop and maintain

● all of the above

Question 22 **1 / 1 point**

A class file that you add to a project has the extension

● cs

○ class

○ cls

○ clss

Question 23 **1 / 1 point**

Which of the variables declared in the following class is an instance variable?

```
public class Customer
{
   public string firstName;
   private static int count;

   public string GetDisplayText()
   {
      string displayText = firstName + count;
      return displayText;
```

```
    }
}
```

● `firstName`

○ `count`

○ `displayText`

Question 24 **1 / 1 point**

A constructor is just a special type of ______________ that instantiates an object from the class.

Answer: method

Question 25 **0 / 1 point**

A ______________ can be used to get and set the value that's stored in a private field.

Answer: auto-implemented properties

Question 26 **0 / 1 point**

Code a statement that creates an instance of the Account class using a constructor that has two parameters named firstName and age, and store the object in a variable named account. Assume that variables with those names have already been declared and initialized so you can pass those variables to the constructor.

____

Answer: Account account = new Account(string firstName, string age);

Question 27 **1 / 1 point**

In a three-layer application, the three layers are

○ the business layer, the middle layer, and the database layer

● the presentation layer, the middle layer, and the database layer

○ the presentation layer, the access layer, and the database layer

○ the presentation layer, the middle layer, and the dataset layer

Question 28 **1 / 1 point**

A read-only property consists of just

○ a get method

○ a set method

○ a set accessor

● a get accessor

Question 29 **0 / 1 point**

When you develop an application using a three-layer architecture, the layer that stores the business rules and classes is typically called the ________________ layer.

Answer: business rules

Question 30 **1 / 1 point**

To begin the declaration for a property, you code the public keyword followed by

○ the name of the property

○ the name of the property and its arguments

● the data type and the name of the property

○ the data type, the name of the property, and the arguments

Question 31 **1 / 1 point**

You can use an auto-implemented property when

○ you don’t need to store the value of the property in an instance variable

● you want the get and set accessors to simply return and set the value of an instance variable

○ you want the instance variable that’s associated with the property to be public

○ you want the property to be private

Chapter 13 Random

Question 32 **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.) 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: private void CustomerListChange(CustomerList customers);

Question 33 **1 / 1 point**

The declaration for an event specifies

● a delegate that will handle the event and the event’s name

○ a class that will handle the event and the event’s name

○ the object that will send the event and the event’s arguments

○ the class that will send the event and the event’s arguments

Question 34 **1 / 1 point**

When coding a business class, why would you want to throw an argument exception from a property or method?

● So the business class is completely self-contained and doesn’t depend on classes that may be coded by other programmers to validate data

○ So the user interface classes don’t have to validate data

○ Because it’s more efficient to throw an exception than to validate data before passing that data to a business class

○ Because it makes your code easier to read

Question 35 **1 / 1 point**

To declare an indexer, you code the ____________ keyword followed by the brackets that contain the parameter that defines the index.

Answer: this

Question 36 **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.) 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 += newCustomer;

Question 37 **1 / 1 point**

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

Answer: !=

Question 38 **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 39 **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 40 **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 41 **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 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]

Question 42 — 1 / 1 point

In the declaration for an overloaded operator, you begin with the public keyword followed by the _____________ keyword.

Answer: static

Question 43 — 0 / 1 point

To overload a binary operator, you need to specify two ___________________ in the declaration.

Answer: operands

Question 44 — 0 / 1 point

Code example 13-2

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

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

○ `Customer`

● `NameChanged`

○ `Customer_NameChanged`

○ `EventHandler`

Question 45 **1 / 1 point**

To raise an event, you code

○ the name of the delegate along with its arguments

○ the name of the event handler

● the name of the event along with its arguments

○ none of the above

Question 46 **0 / 1 point**

Operator overloading is typically used to

● redefine the function of an existing operator for a built-in data type

○ define a new operator for a user-defined data type

○ define or redefine the function of an existing operator for a user-defined data type

○ make an operator unavailable to a user-defined data type

Question 47 **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);

Question 48 **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 49 **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 50 **0 / 1 point**

To overload the == operator, you must override

○ the Equals and GetHashCode methods inherited from the Object class

○ the static Equals and GetHashCode methods of the Object class

○ the Equals and GetHashCode properties inherited from the Object class

● only the Equals method inherited from the Object class

Attempt Score: 37 / 50

Overall Grade (highest attempt)**:** 37 / 50

Done