

Lesson 2

Introduction to Object-Oriented Programming

Knowledge Assessment

Multiple Choice

Circle the letter that corresponds to the best answer.

1. You want to restrict the access for a method to the containing class or to a class that is derived from the containing class. Which access modifier should you use for this method?
 - a. public
 - b. private
 - c. protected**
 - d. internal
2. In a class, you defined a method called Render. This method provides functionality to render bitmap files on the screen. You would like the derived classes to supersede this functionality to support the rendering of additional image formats. You also want the Render method of the derived classes to be executed even if a derived class is cast as the base class. Which keyword should you use with the definition of the Render method in the base class?
 - a. abstract
 - b. virtual**
 - c. new
 - d. overrides
3. You defined a class AdvMath that defines advanced mathematical functionality. You do not want the functionality of this class to be inherited into derived classes. What keyword should you use to define the AdvMath class?
 - a. sealed**
 - b. abstract
 - c. private
 - d. internal
4. You need to provide query functionality to several of your classes. Each class's algorithm for the query will likely be different. Also, not all the classes have an "is-a" relationship with each other. How should you support this functionality?
 - a. Add the query functionality to a base class with public access modifier.
 - b. Have all the classes inherit from an abstract base class and override the base-class method to provide their own query functionality.
 - c. Have all the classes inherit from a base class that provides the query functionality.

d. Create a common interface that is implemented by all the classes.

5. Which of the following class elements should you use to define the behavior of a class?
- a. Method**
 - b. Property
 - c. Event
 - d. Delegate
6. You are writing code for a class named Product. You need to make sure that the data members of the class are initialized to their correct values as soon as you create an object of the Product class. The initialization code should be always executed. What should you do?
- a. Create a static method in the Product class to initialize data members.
 - b. Create a constructor in the Product class to initialize data members.**
 - c. Create a static property in the Product class to initialize data members.
 - d. Create an event in the Product class to initialize data members.
7. You are creating a new class named Square that is derived from the Polygon class. The Polygon class has the following code:

```
class Polygon
{
    public virtual void Draw()
    {
        // additional code...
    }
}
```

The Draw method in the Square class should provide new functionality but also hide the Polygon class implementation of the Draw method. Which code segment should you use to accomplish this?

- a.
- ```
class Square : Polygon
{
 public override void Draw()
 {
 // additional code ...
 }
}
```
- b.**

```
class Square : Polygon
{
 public new void Draw()
 {
 // additional code ...
 }
}
```

c.

```
class Square : Polygon
{
 public virtual void Draw()
 {
 // additional code ...
 }
}
```

d.

```
class Square : Polygon
{
 public static void Draw()
 {
 // additional code ...
 }
}
```

8. You are creating a new class named Rectangle. You write the following code:

```
class Rectangle : IComparable
{
 public double Length { get; set; }
 public double Width { get; set; }

 public double GetArea()
 {
 return Length * Width;
 }

 public int CompareTo(object obj)
 {
 // to be completed
 }
}
```

You need to complete the definition of the CompareTo method to enable comparison of the Rectangle objects. Which of the following code should you write?

a.

```
public int CompareTo(object obj)
{
 Rectangle target = (Rectangle)obj;
 double diff = this.GetArea() - target.GetArea();

 if (diff == 0)
 return 0;
 else if (diff > 0)
 return 1;
 else return -1;
}
```

b.

```
public int CompareTo(object obj)
{
 Rectangle target = (Rectangle)obj;
 double diff = this.GetArea() - target.GetArea();

 if (diff == 0)
 return 1;
 else if (diff > 0)
 return -1;
 else return 0;
}
```

c.

```
public int CompareTo(object obj)
{
 Rectangle target = (Rectangle)obj;

 if (this == target)
 return 0;
 else if (this > target)
 return 1;
 else return -1;
}
```

d.

```
public int CompareTo(object obj)
{
 Rectangle target = (Rectangle)obj;
```

```

 if (this == target)
 return 1;
 else if (this > target)
 return -1;
 else return 0;
 }

```

9. You are writing code for a new method named Process:

```

void Process(object o)
{

}

```

The code receives a parameter of type object. You need to cast this object into the type Rectangle. At times, the value of o that is passed to the method might not be a valid Rectangle value. You need to make sure that the code does not generate any System.InvalidCastException errors while doing the conversions. Which of the following lines of code should you use inside the Process method to accomplish this goal?

- a. Rectangle r = (Rectangle) o;
- b. Rectangle r = o as Rectangle;**
- c. Rectangle r = o is Rectangle;
- d. Rectangle r = (o != null) ? o as rectangle : (Rectangle) o;

10. You are writing code to handle events in your program. You define a delegate named RectangleHandler like this:

```
public delegate void RectangleHandler(Rectangle rect);
```

You also create a variable of the RectangleHandler type as follows:

```
RectangleHandler handler;
```

Later in the program, you need to add a method named DisplayArea to the method invocation list of the handler variable. The signature of the DisplayArea method matches the signature of the RectangleHandler method. Any code that you write should not affect any existing event-handling code. Given this restriction, which of the following code should you write?

- a. handler = new RectangleHandler(DisplayArea);
- b. handler = DisplayArea;
- c. handler += DisplayArea;**
- d. handler -= DisplayArea;

### Fill in the Blank

Complete the following sentences by writing the correct word or words in the blanks provided.

1. A(n) **class** is a blueprint of an object.

2. A class that does not provide a complete implementation must be declared with the keyword **abstract**.
3. Classes that want to support comparison must implement the `Comparable` interface and then provide a body for the **CompareTo** method.
4. You can use the **is** operator to check whether it is legal to cast one type to another type.
5. Three main features of an object-oriented programming language are **encapsulation**, **inheritance**, and **polymorphism**.
6. You can use **namespaces** to group related classes in order to reduce name collisions.
7. The **this** keyword refers to the current instance of a class.
8. A(n) **delegate** is a type that references a method.
9. A(n) **struct** is a value type, whereas a(n) **class** is a reference type.
10. You can use the **static** keyword to declare a member that belongs to the class itself rather than to a specific object.

### Competency Assessment

#### Scenario 2-1: Creating Properties

---

You need to create a class named `Product` that represents a product. The class has a single property named `Name`. Users of the `Product` class should be able to get and set the value of the `Name` property. However, any attempt to set the value of `Name` to an empty string or a null value should raise an exception. Also, users of the `Product` class should not be able to access any other data members of the `Product` class. How will you create such a class?

1. Create a new Visual C# class named `Product.cs`.
2. Replace the code for the `Product` class with the following code:

```
public class Product
{
 private string name;
 public string Name
 {
 get
 {
 return name;
 }
 set
 {
 if (string.IsNullOrEmpty(value))
```

```
 {
 throw new ArgumentException(
 "Name cannot be null or empty string",
 "Name");
 }
 name = value;
 }
}
}
```

**3. Select Build > Build Solution to compile the code.**

### Scenario 2-2: Creating a Struct

---

You are developing a game that needs to represent the location of a target in the three-dimensional space. The location is identified by the three integer values denoted x, y, and z. You will create thousands of these data structures in your program and you need a lightweight, efficient way to store this data in memory. Also, it is unlikely that you will need to inherit any other types from this location type. How should you represent the location in your program?

- 1. Create a new Visual C# class file named Location.cs**
- 2. Replace the code for the Location class with the following code:**

```
public struct Location
{
 public int x, y, z;

 public Location(int x, int y, int z)
 {
 this.x = x;
 this.y = y;
 this.z = z;
 }
}
```

**3. Select Build > Build Solution to compile the code.**

## Proficiency Assessment

### Project 2-1: Overriding the ToString Method

---

You are writing code for a Product class. The Product class contains the name and price of a product. You need to override the base class (System.Object) method ToString to provide information about the objects of the product class to the calling code. What code do you need to write for the Product class in order to meet this requirement?

1. Create a new Visual C# class named Product.cs.
2. Replace the definition of the Product class with the following code:

```
class Product
{
 string name;
 double price;
 Product(string name, double price)
 {
 this.name = name;
 this.price = price;
 }
 public override string ToString()
 {
 string s = price.ToString();
 return "Product: " + name + " " + s;
 }
}
```

3. Select Build > Build Solution to compile the code.

### Project 2-2: Creating and Handling Events

---

You are writing code for creating and handling events in your program. The class SampleClass needs to implement the following interface:

```
public delegate void SampleDelegate();
public interface ISampleEvents
{
 event SampleDelegate SampleEvent;
 void Invoke();
}
```



You need to write code for the SampleClass and for a test method that creates an instance of the SampleClass and invokes the event. What code should you write?

1. Create a new Visual C# class named SampleClass.cs, then replace the code for the Sample class with the following code:

```
public class SampleClass : ISampleEvents
{
 public event SampleDelegate SampleEvent;
 public void Invoke()
 {
 if (SampleEvent != null)
 SampleEvent();
 }
}
```

2. Add another class named TestClass to the file:

```
public class TestClass
{
 static private void TestHandler()
 {
 Console.WriteLine(
"TestHandler is called when the event fires.");
 }

 static public void Main()
 {
 ISampleEvents se = new SampleClass();

 se.SampleEvent += new SampleDelegate(TestHandler);
 se.Invoke();
 }
}
```

3. Build the code and run the project to verify that the event handler is executed.