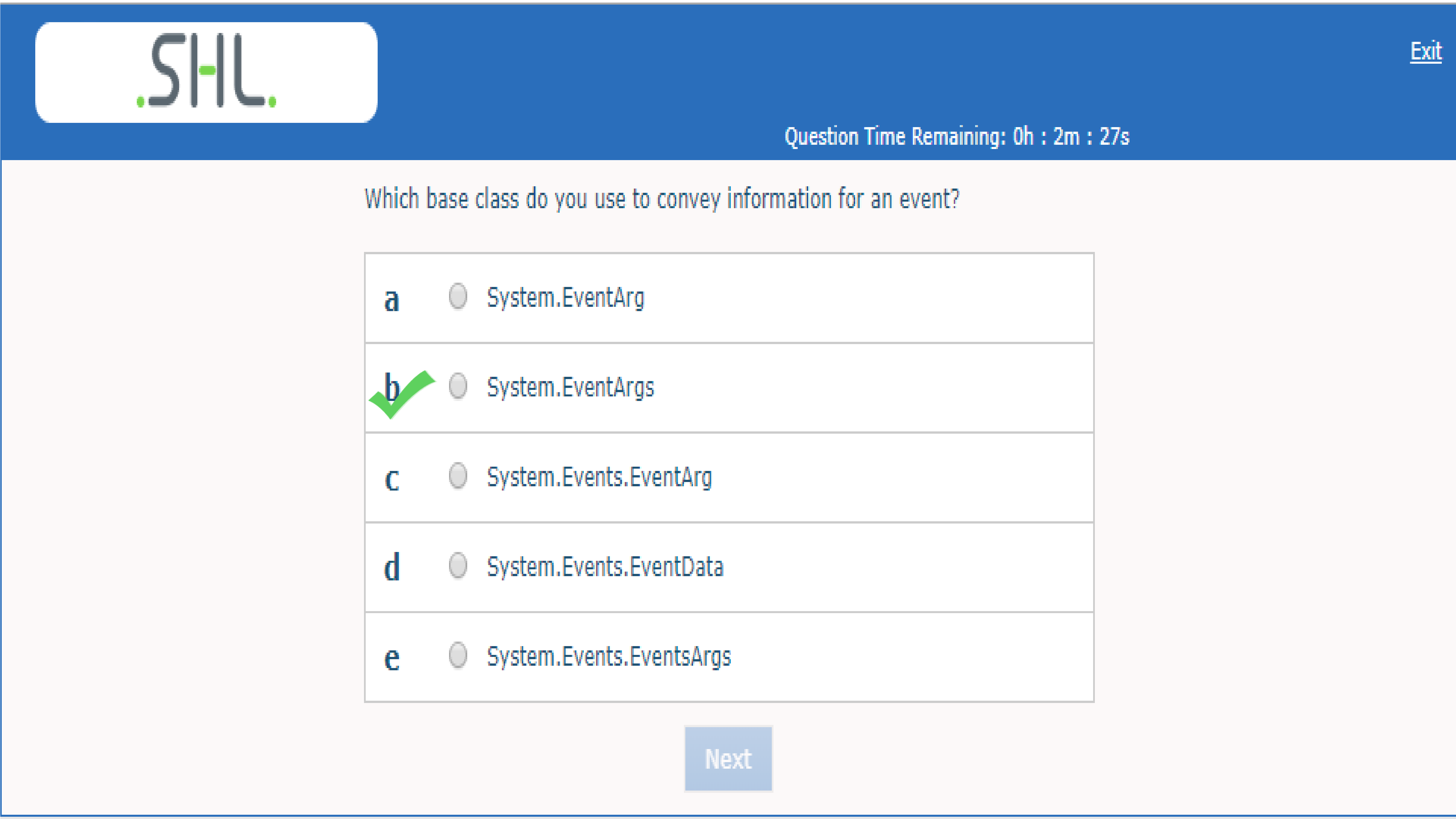


SHL

Exit

Question Time Remaining: 0h : 2m : 27s

Which base class do you use to convey information for an event?

a System.EventArg

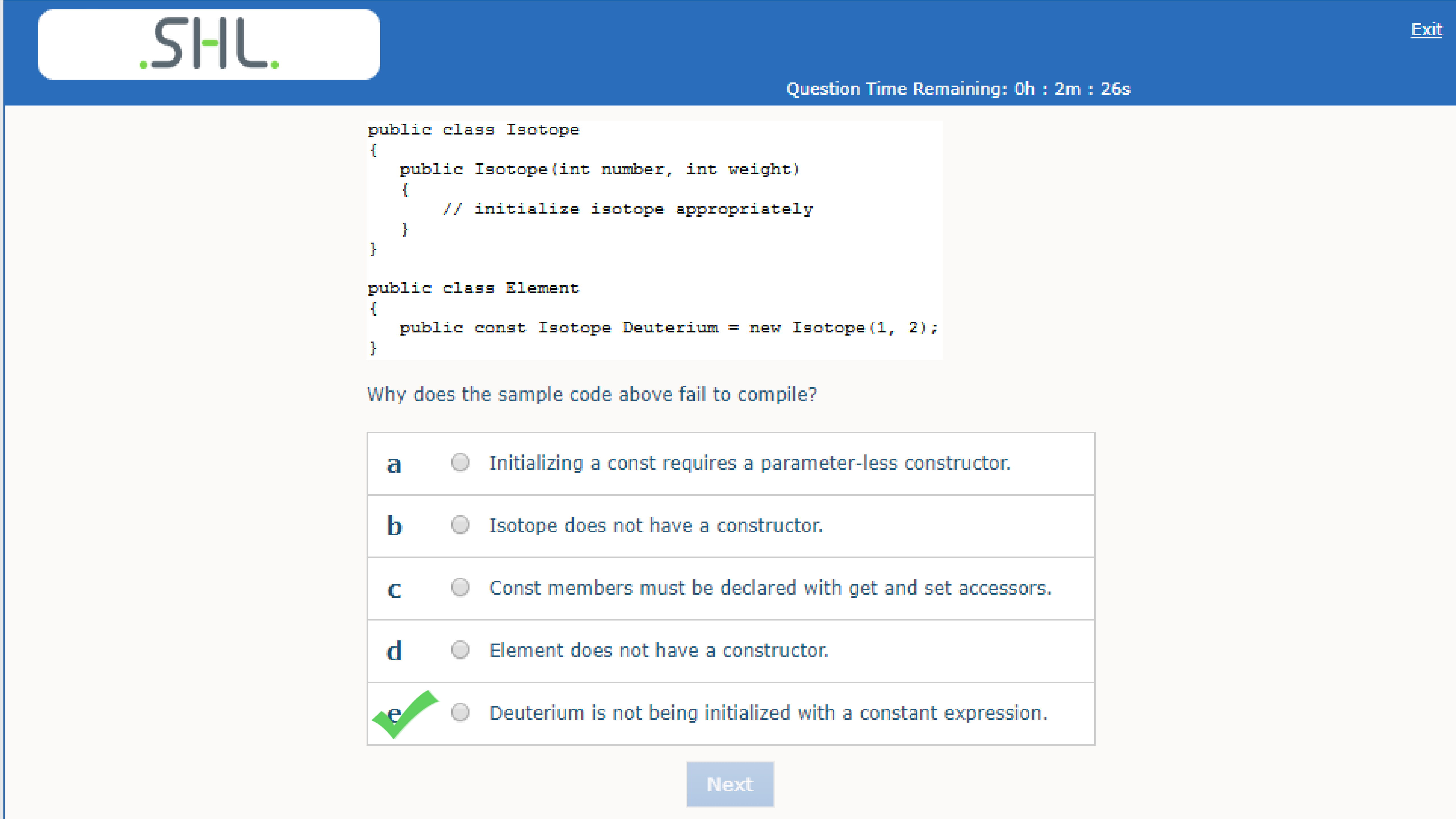
[✓ b System.EventArgs]

c System.Events.EventArg

d System.Events.EventData

e System.Events.EventsArgs

Next



Exit

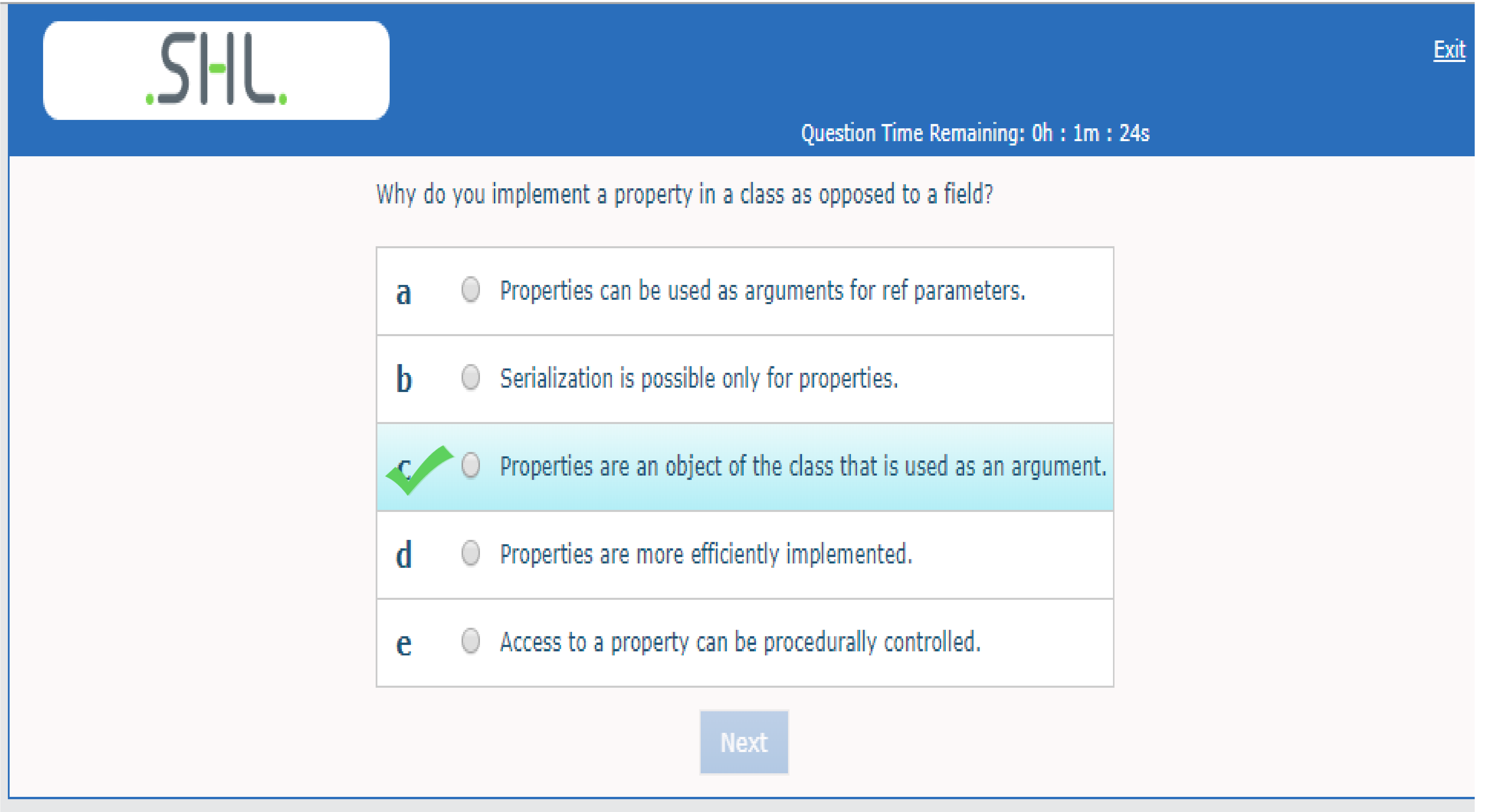
Question Time Remaining: 0h : 2m : 26s

```
public class Isotope
{
   public Isotope(int number, int weight)
   {
       // initialize isotope appropriately
   }
}

public class Element
{
   public const Isotope Deuterium = new Isotope(1, 2);
}
```

Why does the sample code above fail to compile?

a	Initializing a const requires a parameter-less constructor.
b	Isotope does not have a constructor.
c	Const members must be declared with get and set accessors.
d	Element does not have a constructor.
[✓ e]	Deuterium is not being initialized with a constant expression.

Next



online.eu/default?action=testquestion&testaction=-1&timetaken=65565&audiopage=false&dbgs=15401209

Exit

Question Time Remaining: 0h : 1m : 24s

Why do you implement a property in a class as opposed to a field?

a Properties can be used as arguments for ref parameters.

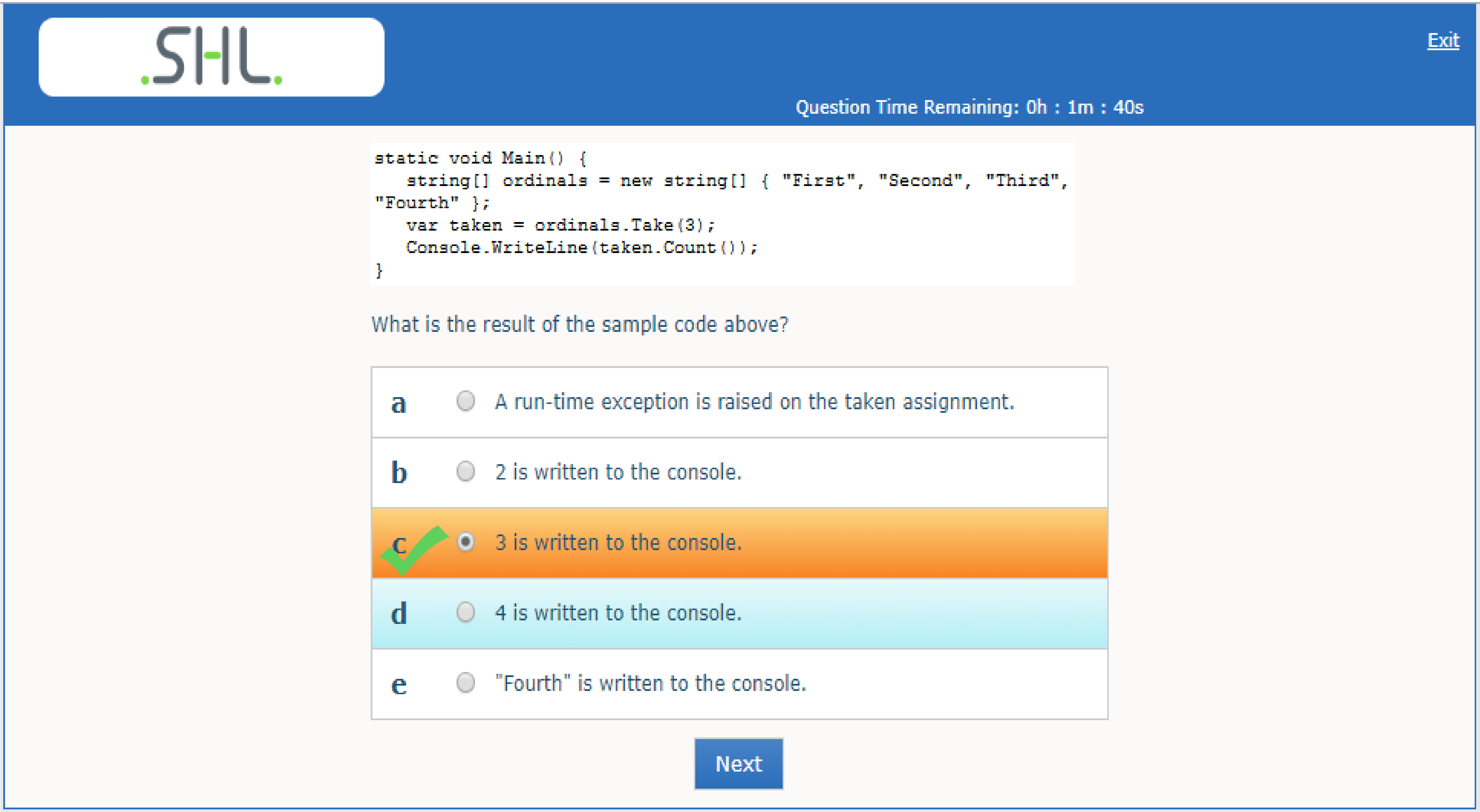
b Serialization is possible only for properties.

[✓ c] Properties are an object of the class that is used as an argument.

d Properties are more efficiently implemented.

e Access to a property can be procedurally controlled.

Next


SHL

Exit

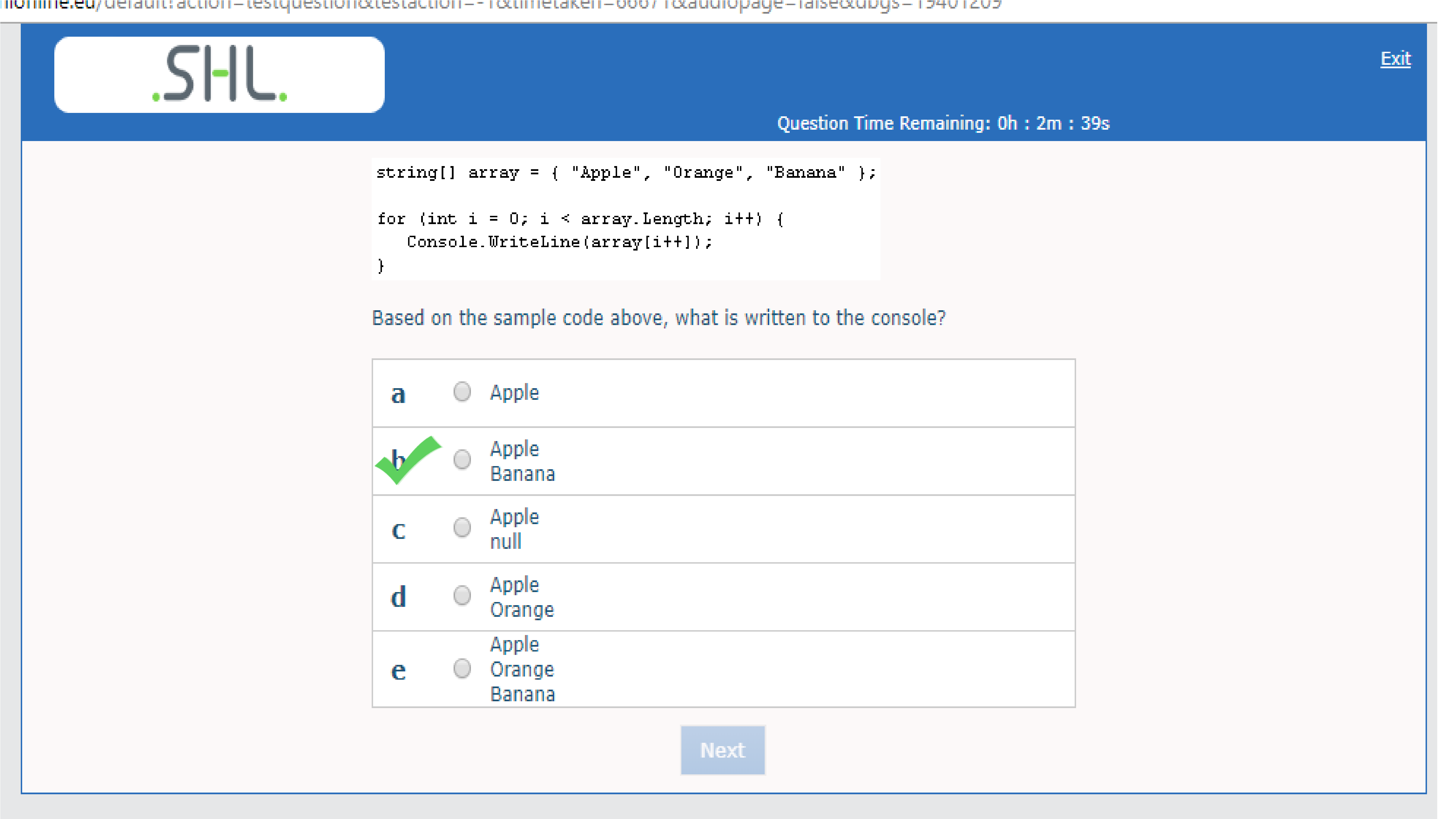
Question Time Remaining: 0h : 1m : 40s

```
static void Main() {
   string[] ordinals = new string[] { "First", "Second", "Third",
"Fourth" };
   var taken = ordinals.Take(3);
   Console.WriteLine(taken.Count());
}
```

What is the result of the sample code above?

a	A run-time exception is raised on the taken assignment.
b	2 is written to the console.
c	[✓ 3 is written to the console.]
d	4 is written to the console.
e	"Fourth" is written to the console.

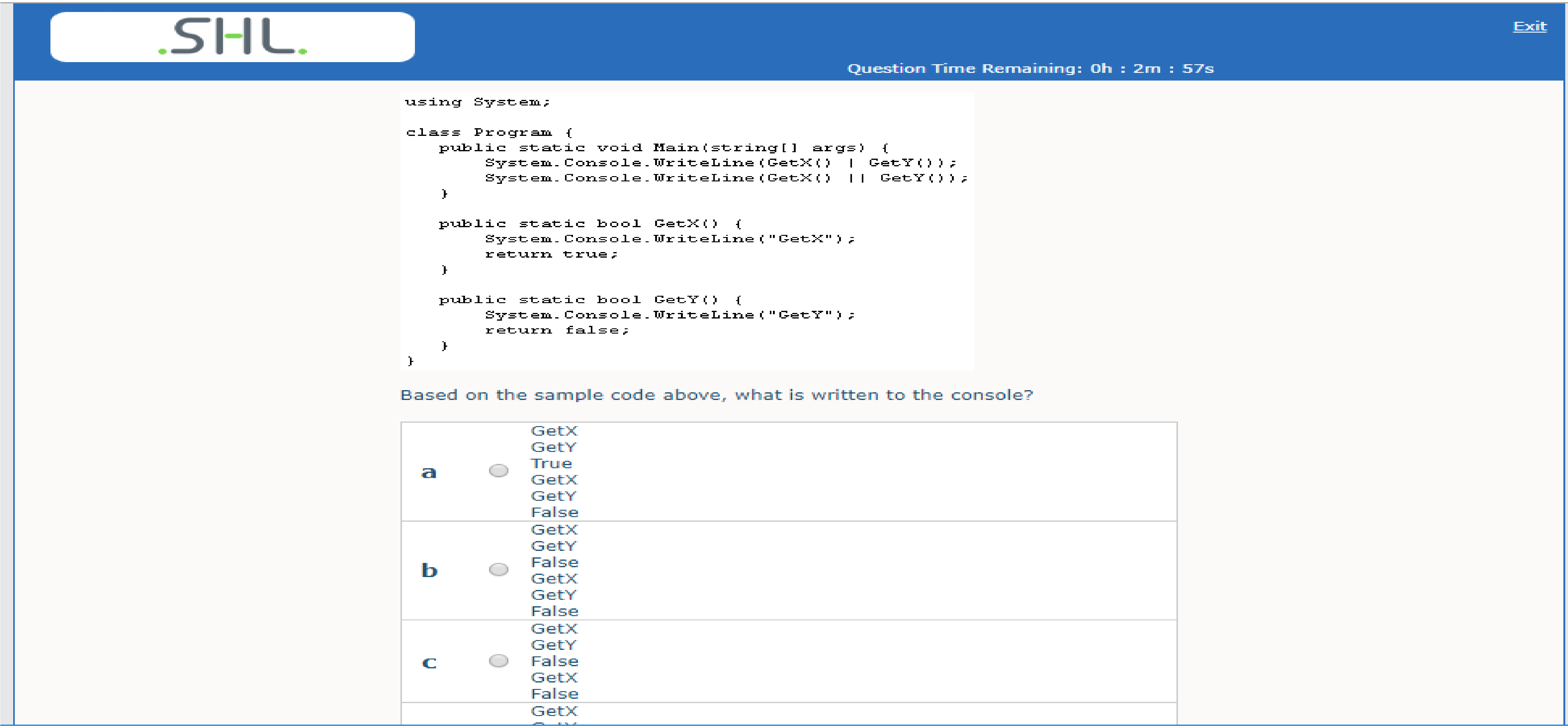
Next


lonline.eu/default?action=testquestion&testaction=-1&timetaken=66671&audiopage=false&dbgs=19401209

```
string[] array = { "Apple", "Orange", "Banana" };

for (int i = 0; i < array.Length; i++) {
   Console.WriteLine(array[i++]);
}
```

Based on the sample code above, what is written to the console?

a	Apple
[✓ b]	Apple Banana
c	Apple null
d	Apple Orange
e	Apple Orange Banana



Question Time Remaining: 0h : 2m : 57s

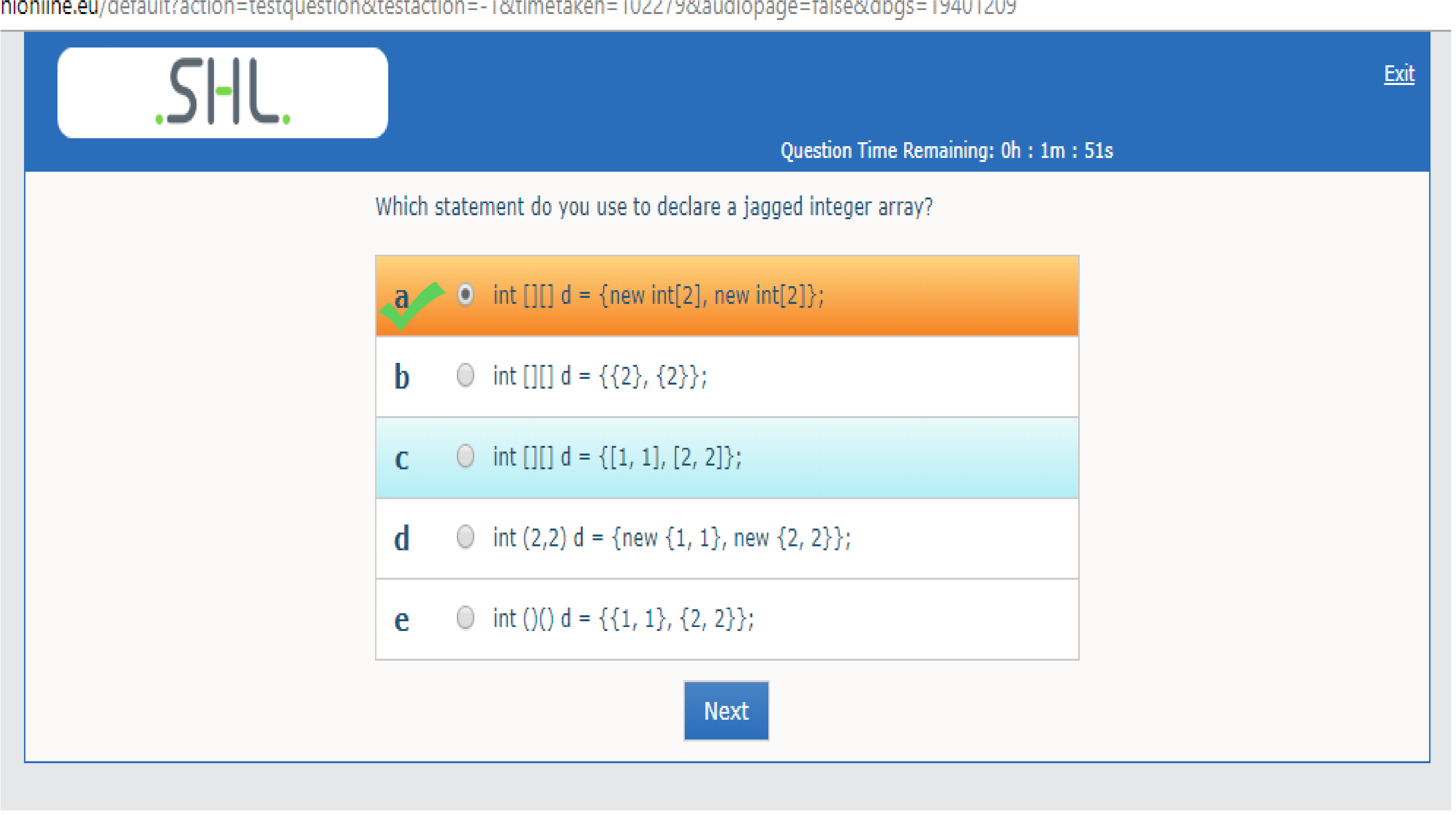
```
using System;

class Program {
    public static void Main(string[] args) {
         System.Console.WriteLine(GetX() | GetY());
         System.Console.WriteLine(GetX() || GetY());
    }

    public static bool GetX() {
         System.Console.WriteLine("GetX");
         return true;
    }

    public static bool GetY() {
         System.Console.WriteLine("GetY");
         return false;
    }
}
```

Based on the sample code above, what is written to the console?

a	○	GetX GetY True GetX GetY False
b	○	GetX GetY False GetX GetY False
c	○	GetX GetY False GetX False
		GetX

[✓ d]

GetX
GetY
True
GetX
True


nlonline.eu/default?action=testquestion&testaction=-1&timetaken=102279&audiopage=false&dbgs=19401209

Which statement do you use to declare a jagged integer array?

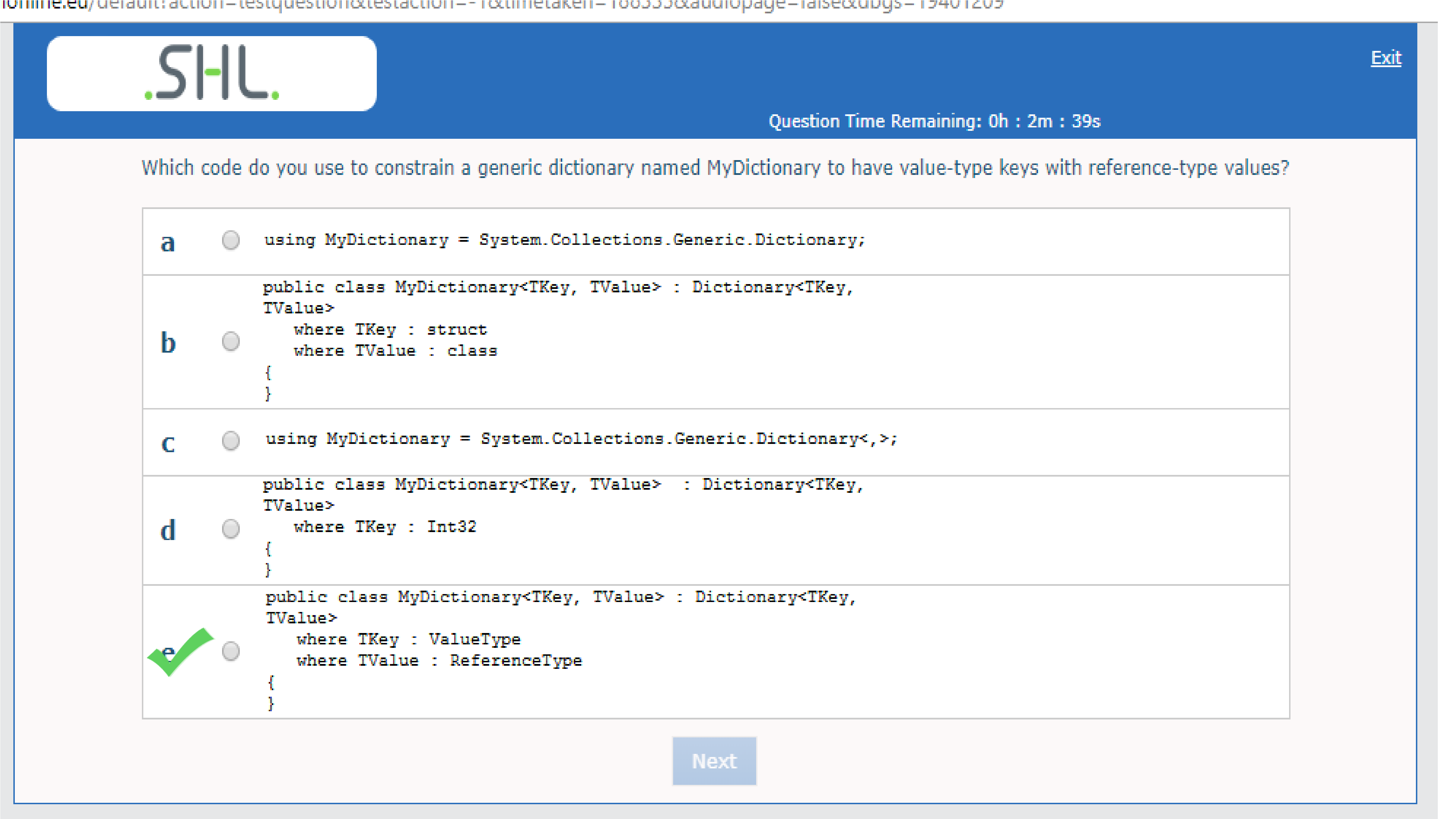
[✓ a] ● int [][] d = {new int[2], new int[2]};

b ○ int [][] d = {{2}, {2}};

c ○ int [][] d = {[1, 1], [2, 2]};

d ○ int (2,2) d = {new {1, 1}, new {2, 2}};

e ○ int ()() d = {{1, 1}, {2, 2}};



ionline.eu/default?action=testquestion&testaction=-1&timetaken=186335&audiopage=false&dbgs=19401209

.SHL.

Exit

Question Time Remaining: 0h : 2m : 39s

Which code do you use to constrain a generic dictionary named MyDictionary to have value-type keys with reference-type values?

a

```
using MyDictionary = System.Collections.Generic.Dictionary;
```

b

```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : struct
   where TValue : class
{
}
```

c

```
using MyDictionary = System.Collections.Generic.Dictionary<,>;
```

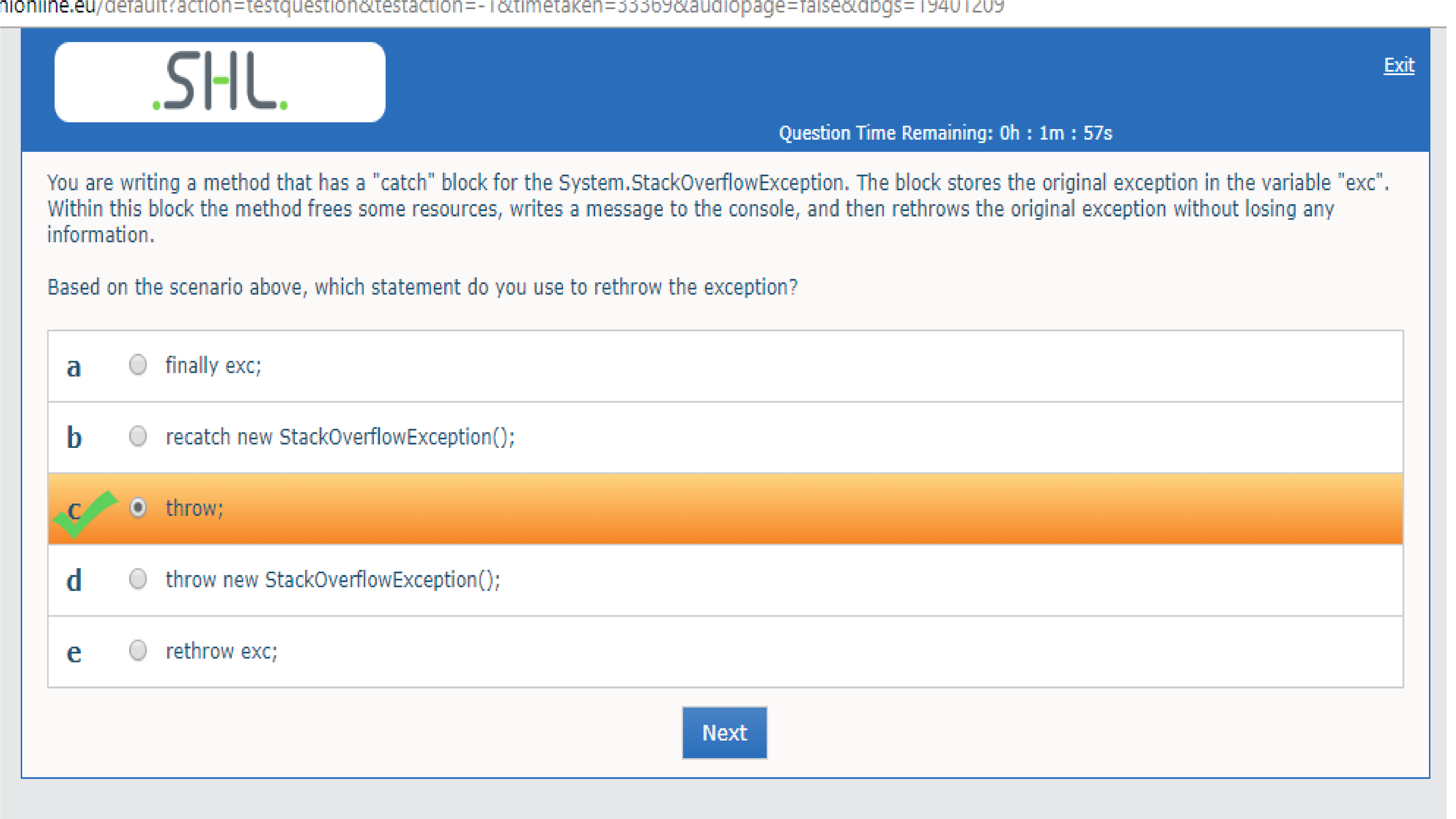
d

```
public class MyDictionary<TKey, TValue>  : Dictionary<TKey,
TValue>
   where TKey : Int32
{
}
```

✓ e

```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : ValueType
   where TValue : ReferenceType
{
}
```

Next



nlonline.eu/default?action=testquestion&testaction=-1&timetaken=33369&audiopage=false&dbgs=19401209

Exit

Question Time Remaining: 0h : 1m : 57s

You are writing a method that has a "catch" block for the System.StackOverflowException. The block stores the original exception in the variable "exc". Within this block the method frees some resources, writes a message to the console, and then rethrows the original exception without losing any information.

Based on the scenario above, which statement do you use to rethrow the exception?

a finally exc;

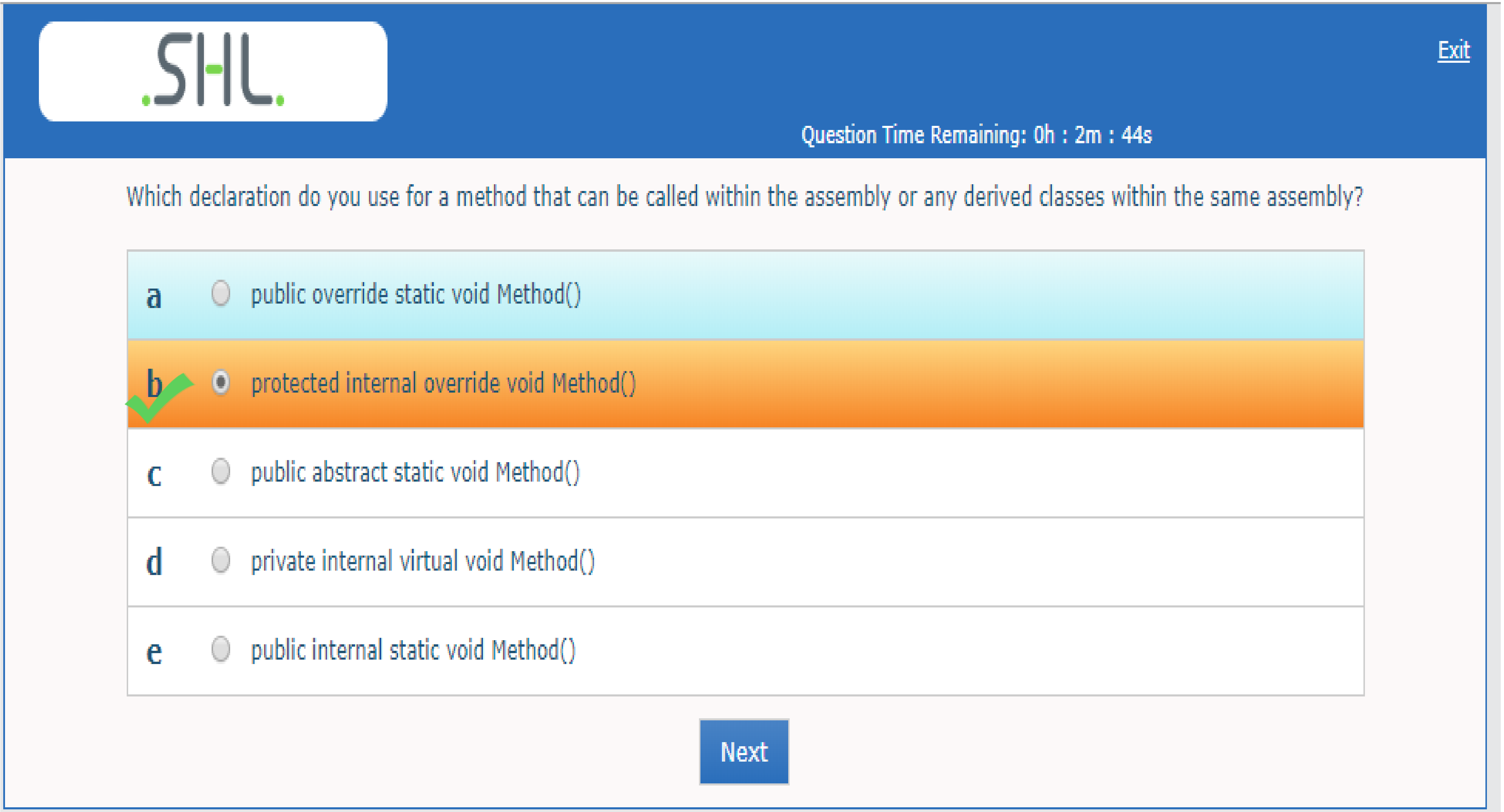
b recatch new StackOverflowException();

[✓ c throw;]

d throw new StackOverflowException();

e rethrow exc;

Next



Question Time Remaining: 0h : 2m : 44s

Which declaration do you use for a method that can be called within the assembly or any derived classes within the same assembly?

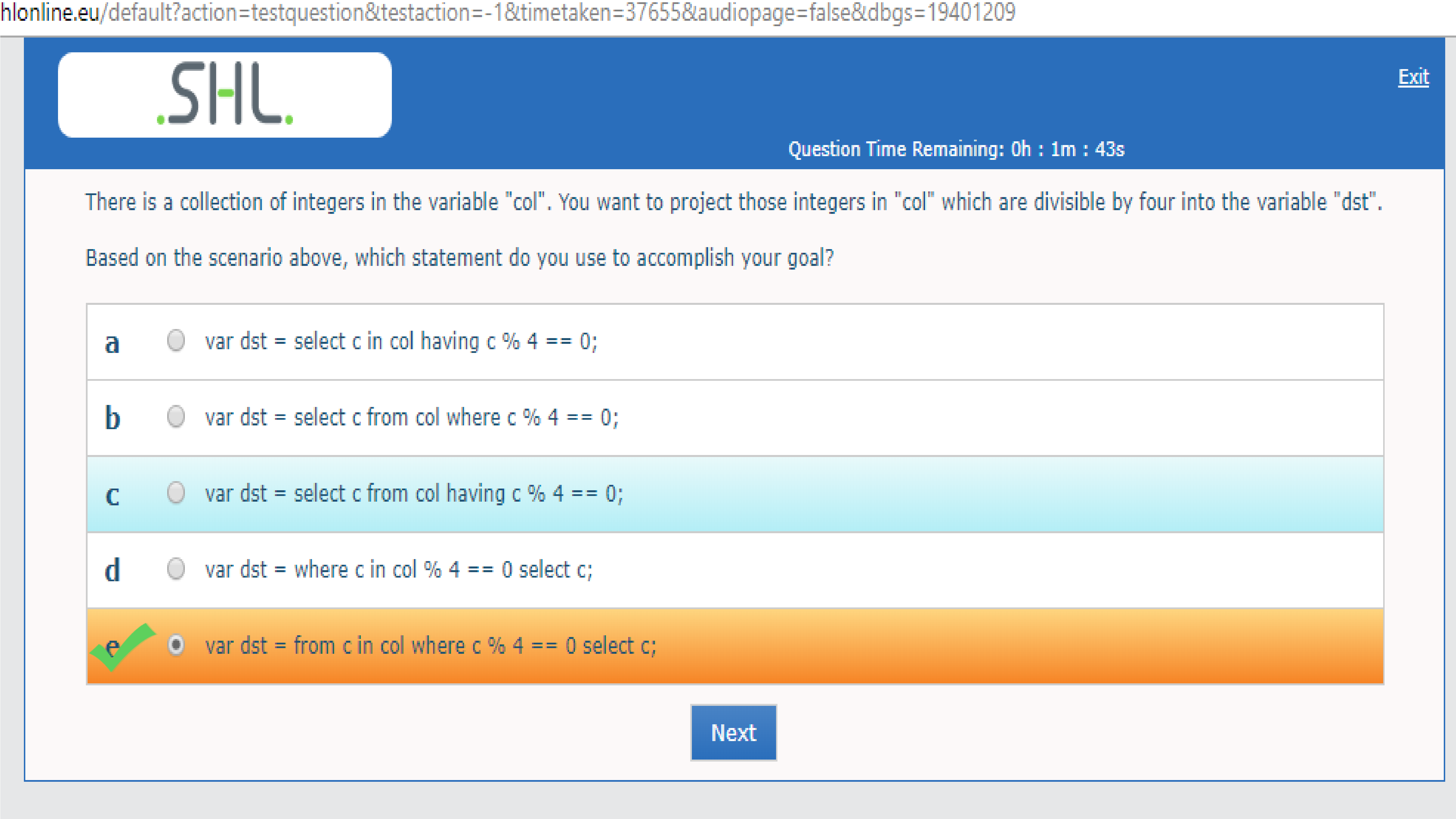
a public override static void Method()

b [✓ protected internal override void Method()]

c public abstract static void Method()

d private internal virtual void Method()

e public internal static void Method()


hlonline.eu/default?action=testquestion&testaction=-1&timetaken=37655&audiopage=false&dbgs=19401209

Exit

Question Time Remaining: 0h : 1m : 43s

There is a collection of integers in the variable "col". You want to project those integers in "col" which are divisible by four into the variable "dst".

Based on the scenario above, which statement do you use to accomplish your goal?

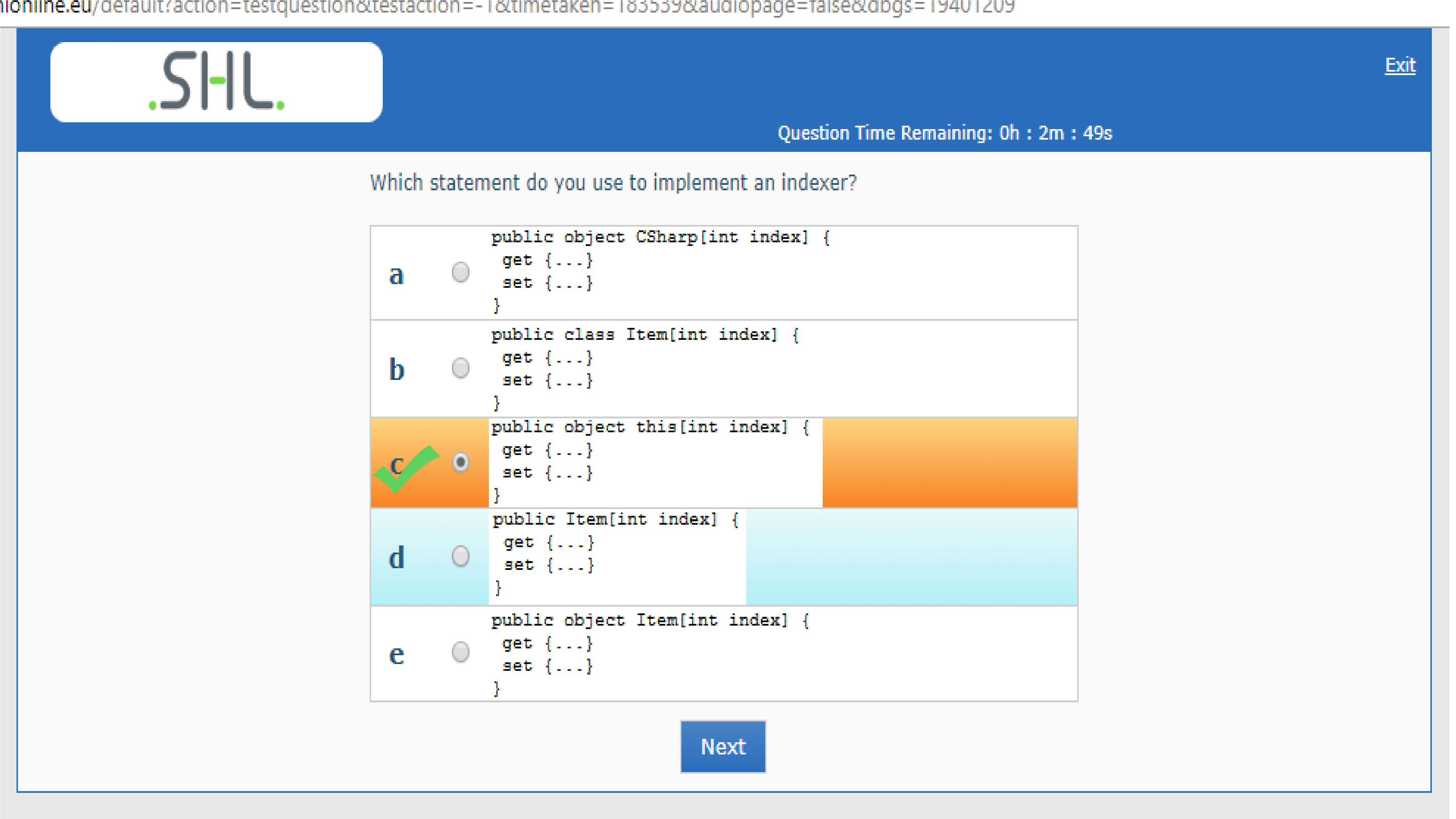
a ○ var dst = select c in col having c % 4 == 0;

b ○ var dst = select c from col where c % 4 == 0;

c ○ var dst = select c from col having c % 4 == 0;

d ○ var dst = where c in col % 4 == 0 select c;

[✓ e ● var dst = from c in col where c % 4 == 0 select c;]



ionline.eu/default?action=testquestion&testaction=-1&timetaken=183539&audiopage=false&dbgs=19401209

Question Time Remaining: 0h : 2m : 49s

Which statement do you use to implement an indexer?

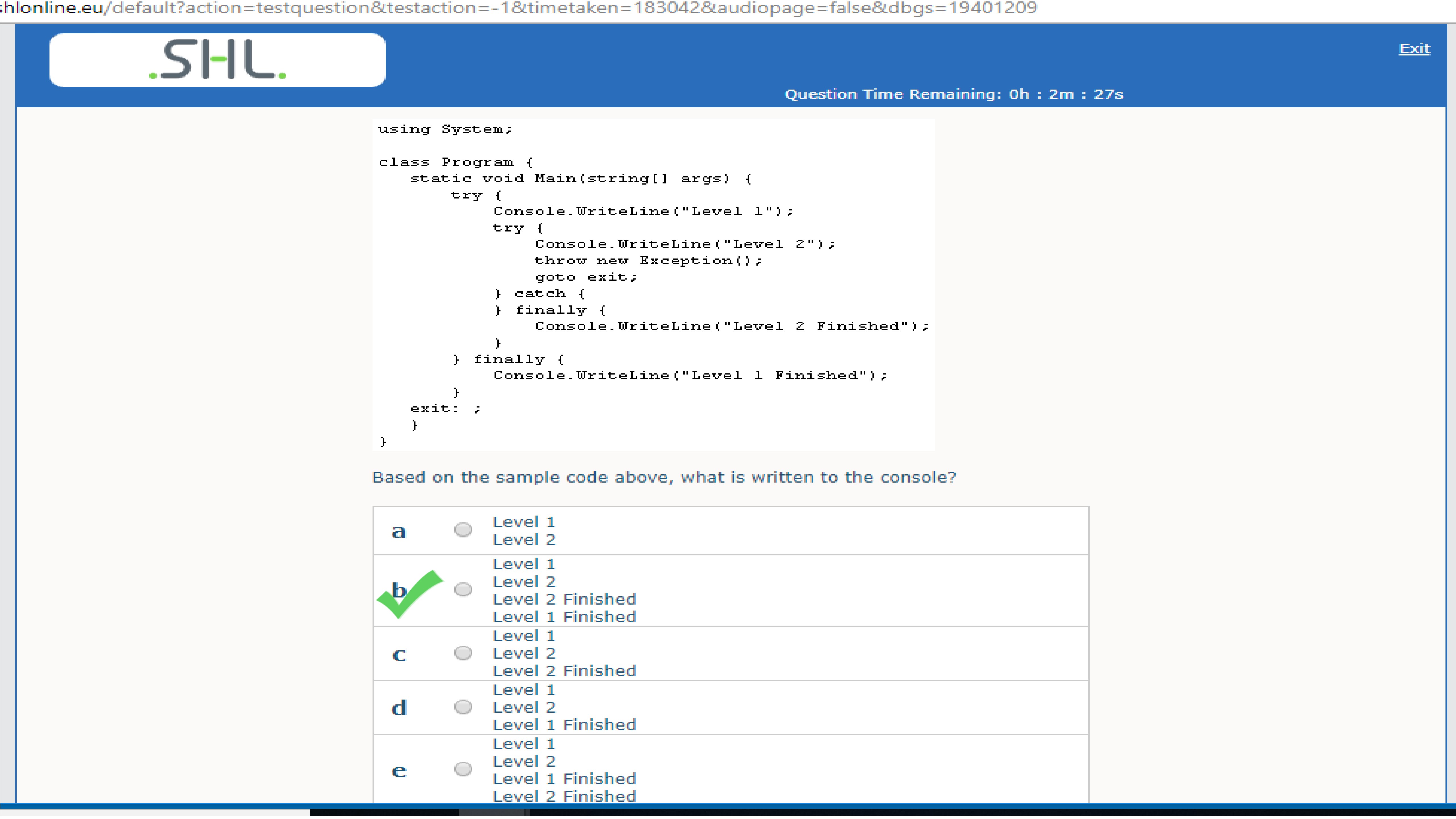
a
```
public object CSharp[int index] {
 get {...}
 set {...}
}
```

b
```
public class Item[int index] {
 get {...}
 set {...}
}
```

[✓ c]
```
public object this[int index] {
 get {...}
 set {...}
}
```

d
```
public Item[int index] {
 get {...}
 set {...}
}
```

e
```
public object Item[int index] {
 get {...}
 set {...}
}
```



hlonline.eu/default?action=testquestion&testaction=-1&timetaken=183042&audiopage=false&dbgs=19401209

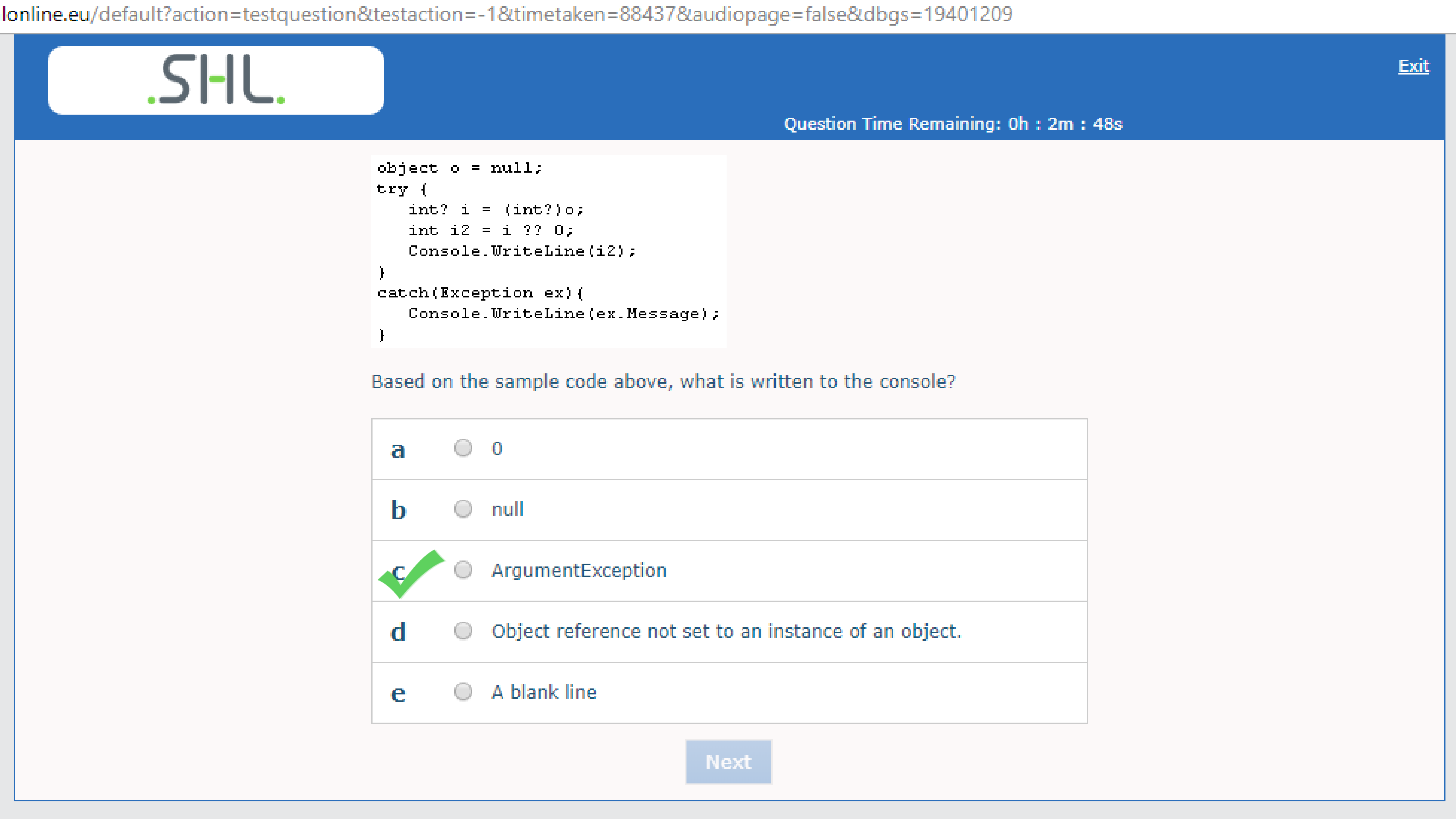
Exit

Question Time Remaining: 0h : 2m : 27s

```
using System;

class Program {
   static void Main(string[] args) {
       try {
           Console.WriteLine("Level 1");
           try {
               Console.WriteLine("Level 2");
               throw new Exception();
               goto exit;
           } catch {
           } finally {
               Console.WriteLine("Level 2 Finished");
           }
       } finally {
           Console.WriteLine("Level 1 Finished");
       }
   exit: ;
   }
}
```

Based on the sample code above, what is written to the console?

a	Level 1 Level 2
[✓ b]	Level 1 Level 2 Level 2 Finished Level 1 Finished
c	Level 1 Level 2 Level 2 Finished
d	Level 1 Level 2 Level 1 Finished
e	Level 1 Level 2 Level 1 Finished Level 2 Finished



lonline.eu/default?action=testquestion&testaction=-1&timetaken=88437&audiopage=false&dbgs=19401209

Exit

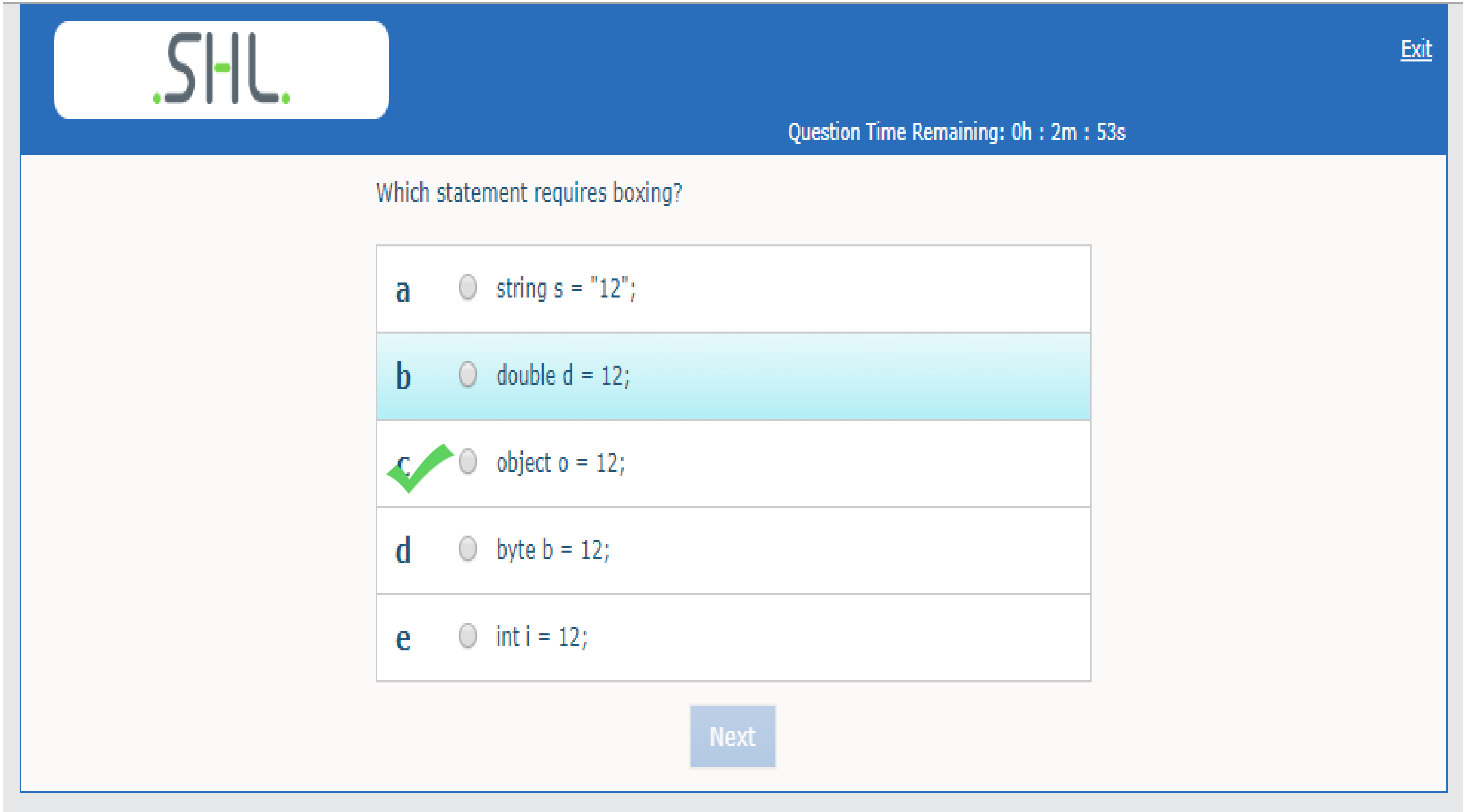
Question Time Remaining: 0h : 2m : 48s

```
object o = null;
try {
    int? i = (int?)o;
    int i2 = i ?? 0;
    Console.WriteLine(i2);
}
catch(Exception ex){
    Console.WriteLine(ex.Message);
}
```

Based on the sample code above, what is written to the console?

a	0
b	null
✓ c	ArgumentException
d	Object reference not set to an instance of an object.
e	A blank line

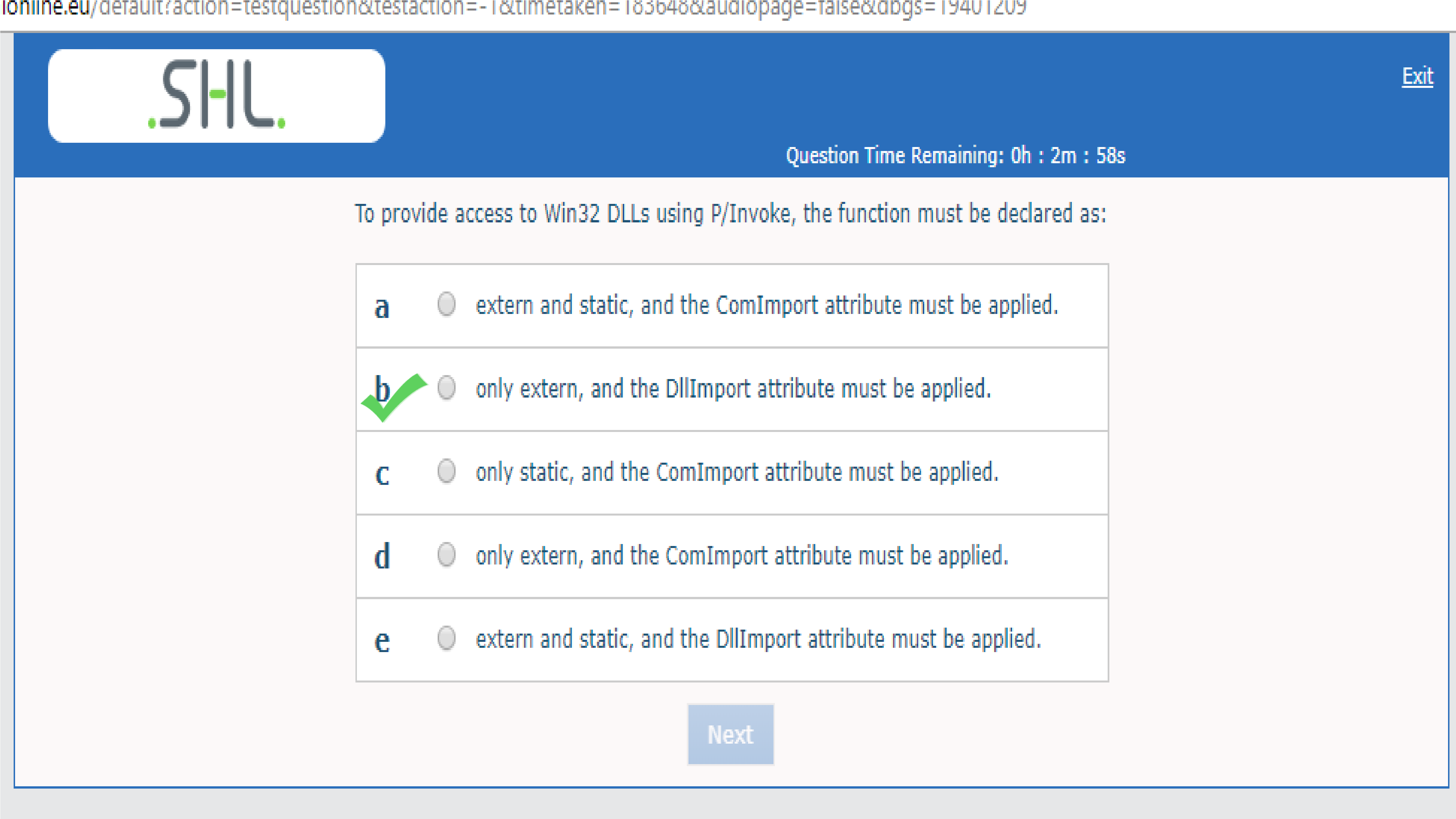
Next



Question Time Remaining: 0h : 2m : 53s

Which statement requires boxing?

- a string s = "12";
- b double d = 12;
- [✓ c] object o = 12;
- d byte b = 12;
- e int i = 12;



ionline.eu/default?action=testquestion&testaction=-1&timetaken=183648&audiopage=false&dbgs=19401209

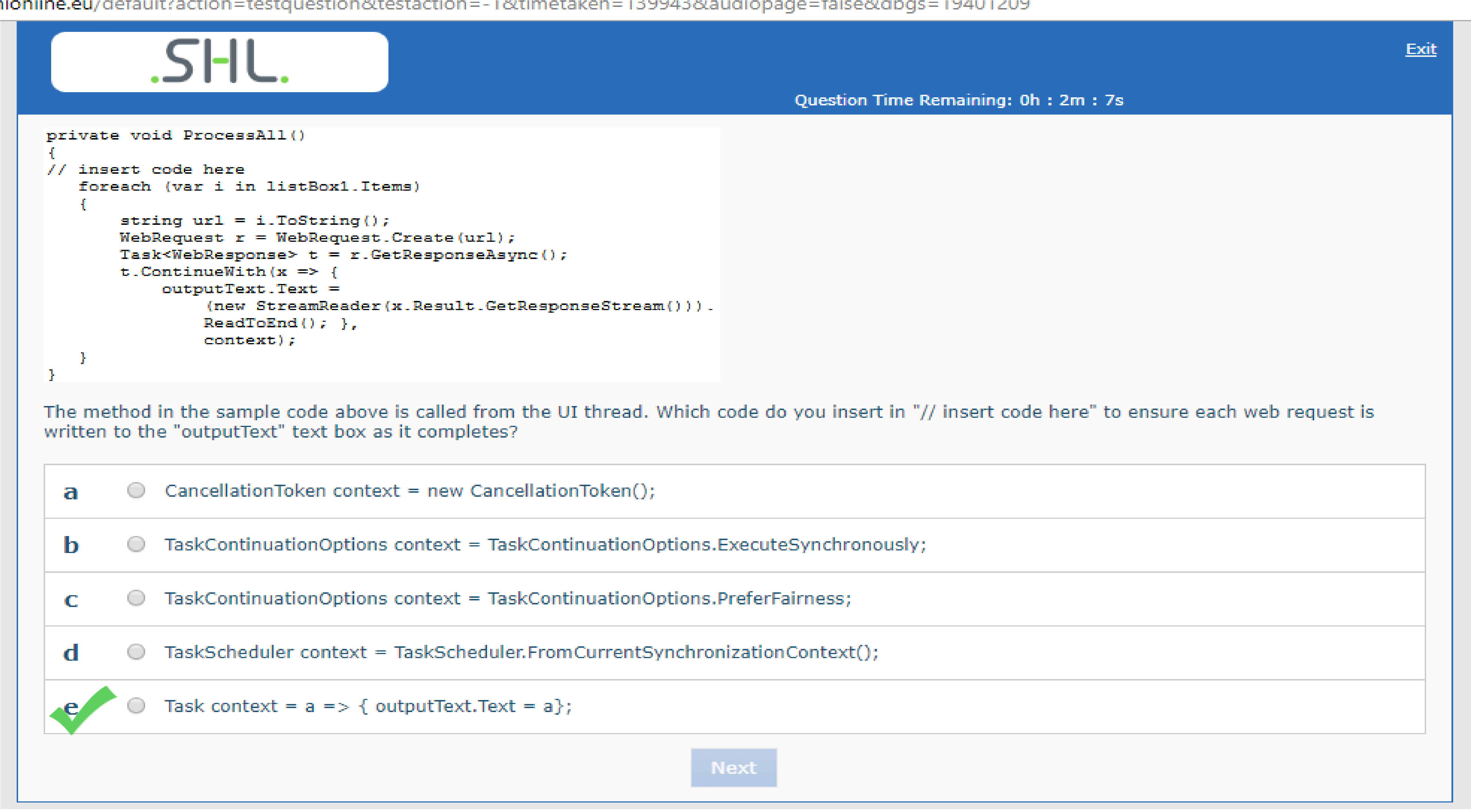
Exit

Question Time Remaining: 0h : 2m : 58s

To provide access to Win32 DLLs using P/Invoke, the function must be declared as:

a	extern and static, and the ComImport attribute must be applied.
[✓ b]	only extern, and the DllImport attribute must be applied.
c	only static, and the ComImport attribute must be applied.
d	only extern, and the ComImport attribute must be applied.
e	extern and static, and the DllImport attribute must be applied.

Next

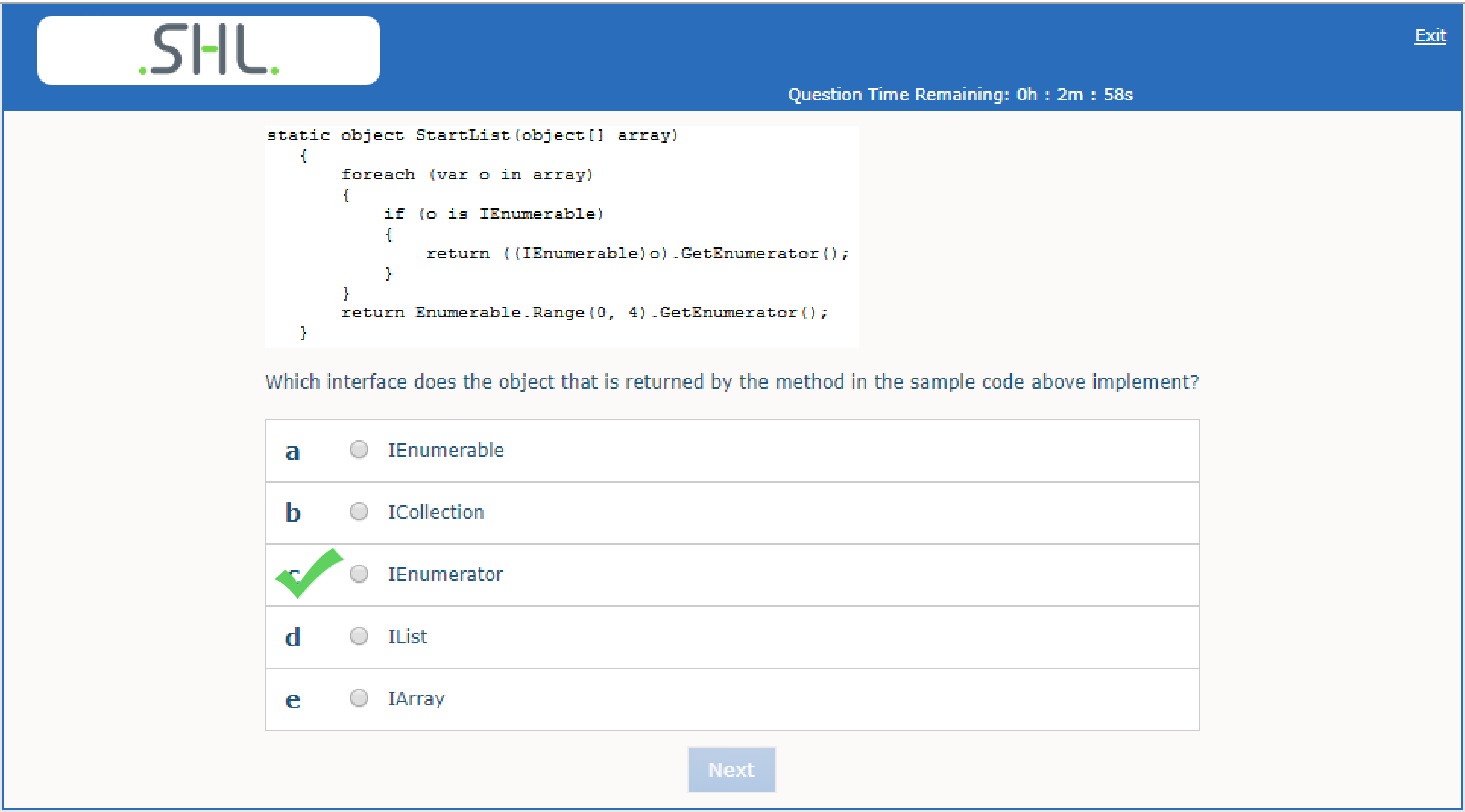


nonline.eu/default?action=testquestion&testaction=-1&timetaken=139943&audiopage=false&dbgs=19401209

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

a	CancellationToken context = new CancellationToken();
b	TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;
c	TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;
d	TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();
[✓ e]	Task context = a => { outputText.Text = a};



hlonline.eu/default?action=testquestion&testaction=-1&timetaken=151286&audiopage=false&dbgs=19401209

Exit

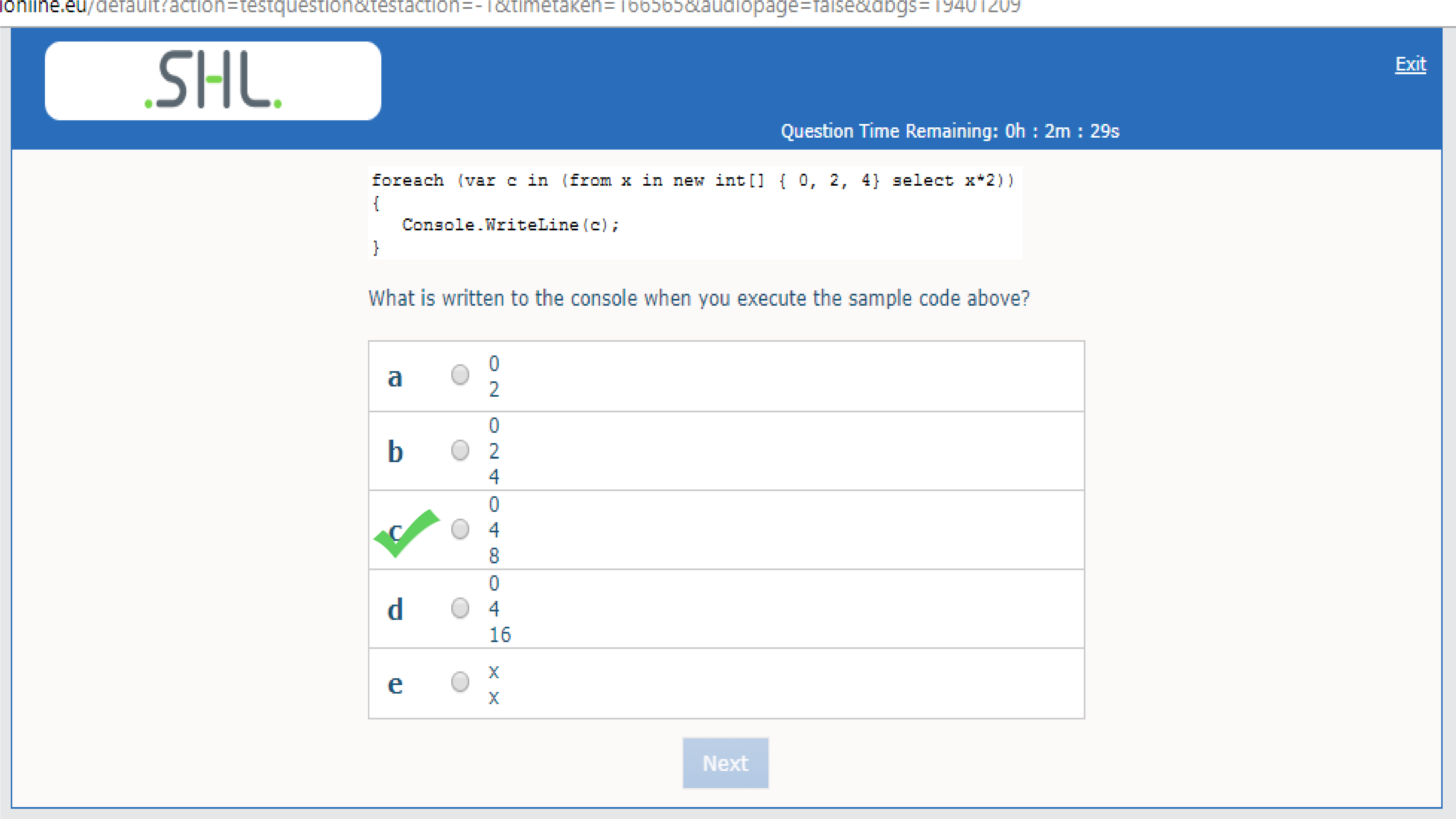
Question Time Remaining: 0h : 2m : 58s

```
static object StartList(object[] array)
    {
        foreach (var o in array)
        {
            if (o is IEnumerable)
            {
                return ((IEnumerable)o).GetEnumerator();
            }
        }
        return Enumerable.Range(0, 4).GetEnumerator();
    }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	IEnumerable
b	ICollection
[✓ c]	IEnumerator
d	IList
e	IArray

Next



ionline.eu/default?action=testquestion&testaction=-1&timetaken=166565&audiopage=false&dbgs=19401209

Exit

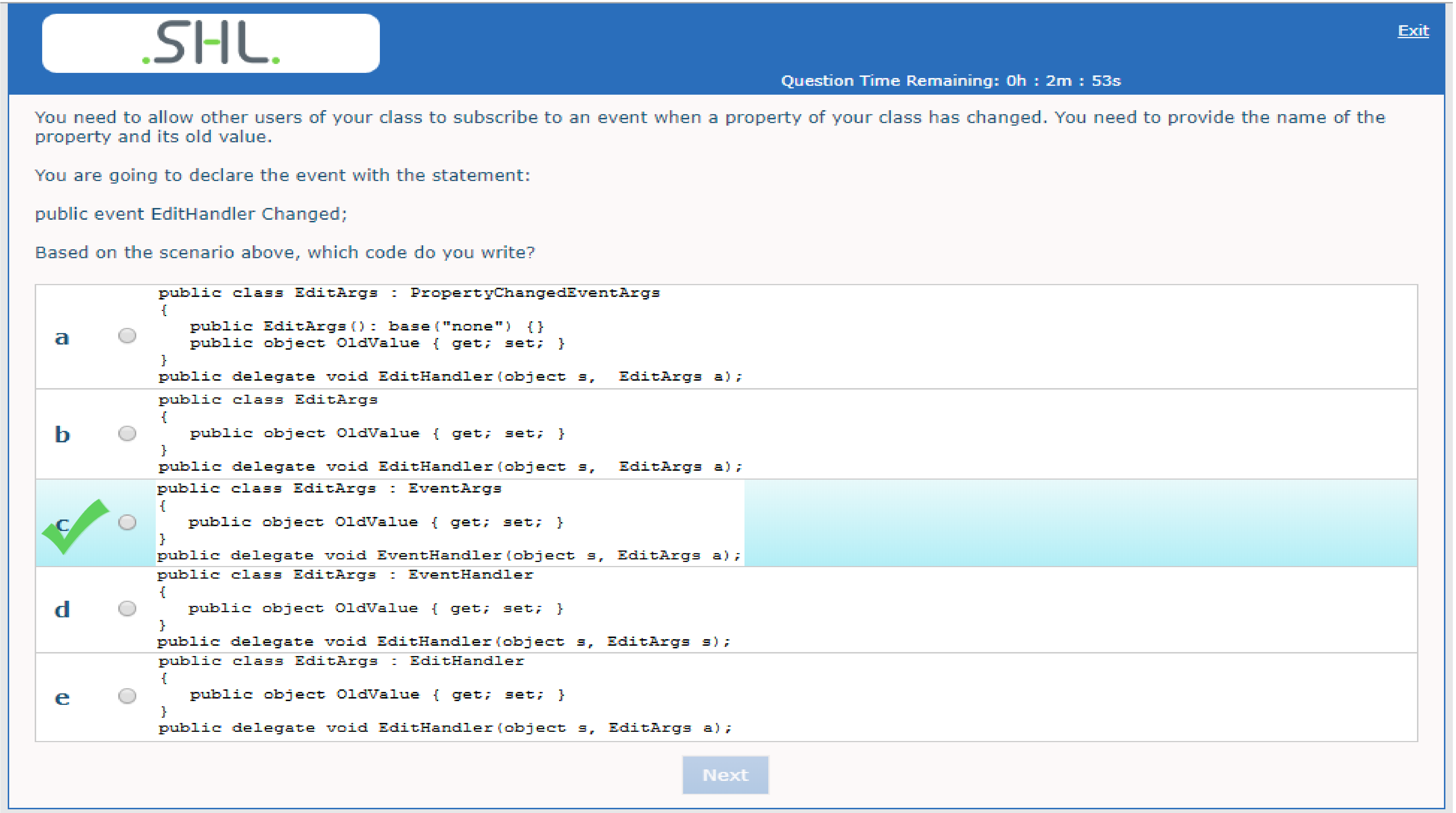
Question Time Remaining: 0h : 2m : 29s

```
foreach (var c in (from x in new int[] { 0, 2, 4} select x*2))
{
   Console.WriteLine(c);
}
```

What is written to the console when you execute the sample code above?

a	0 2
b	0 2 4
✓ c	0 4 8
d	0 4 16
e	x x

Next



Question Time Remaining: 0h : 2m : 53s

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

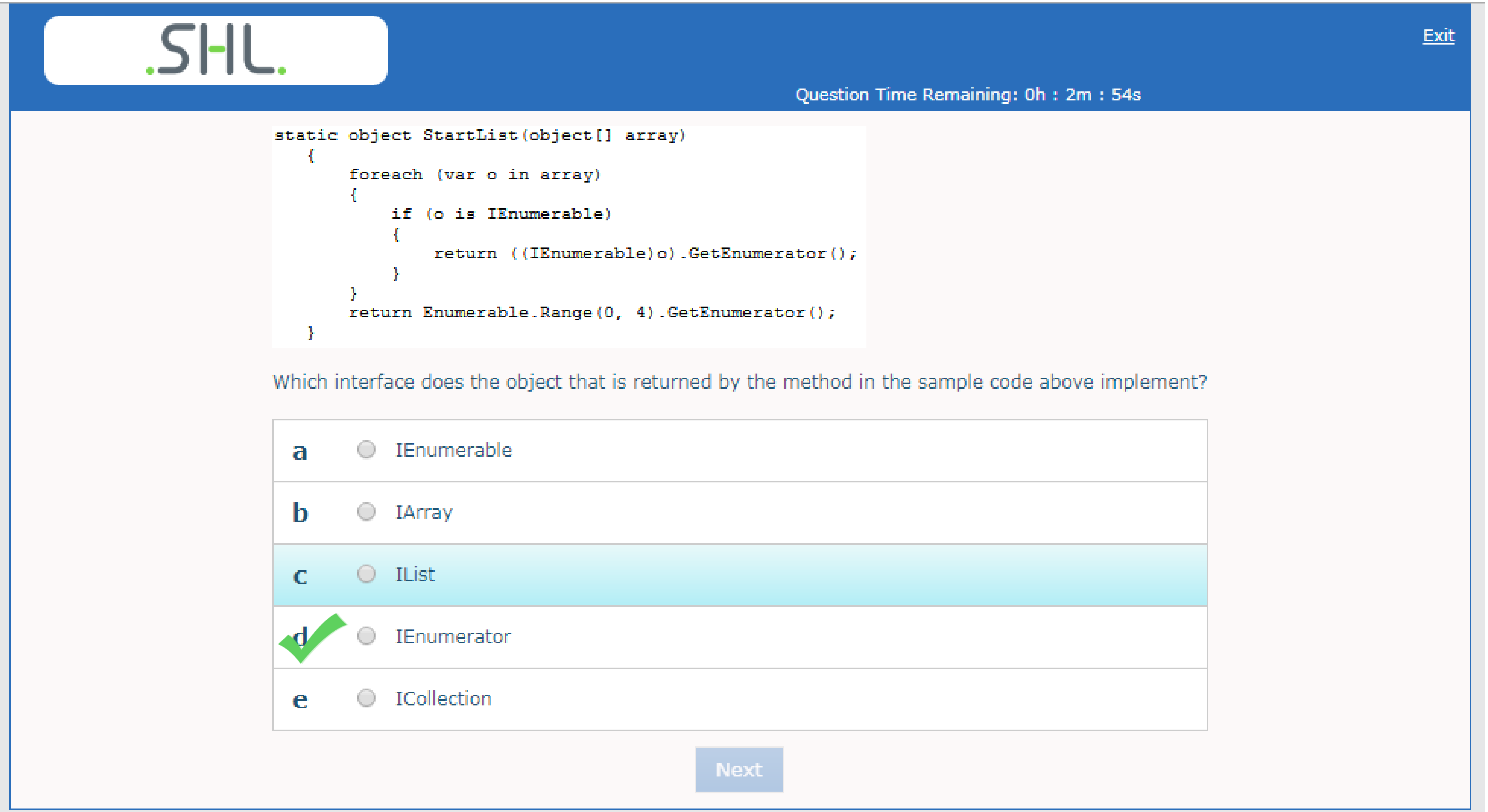
a
```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

b
```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

[✓ c]
```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

d
```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```

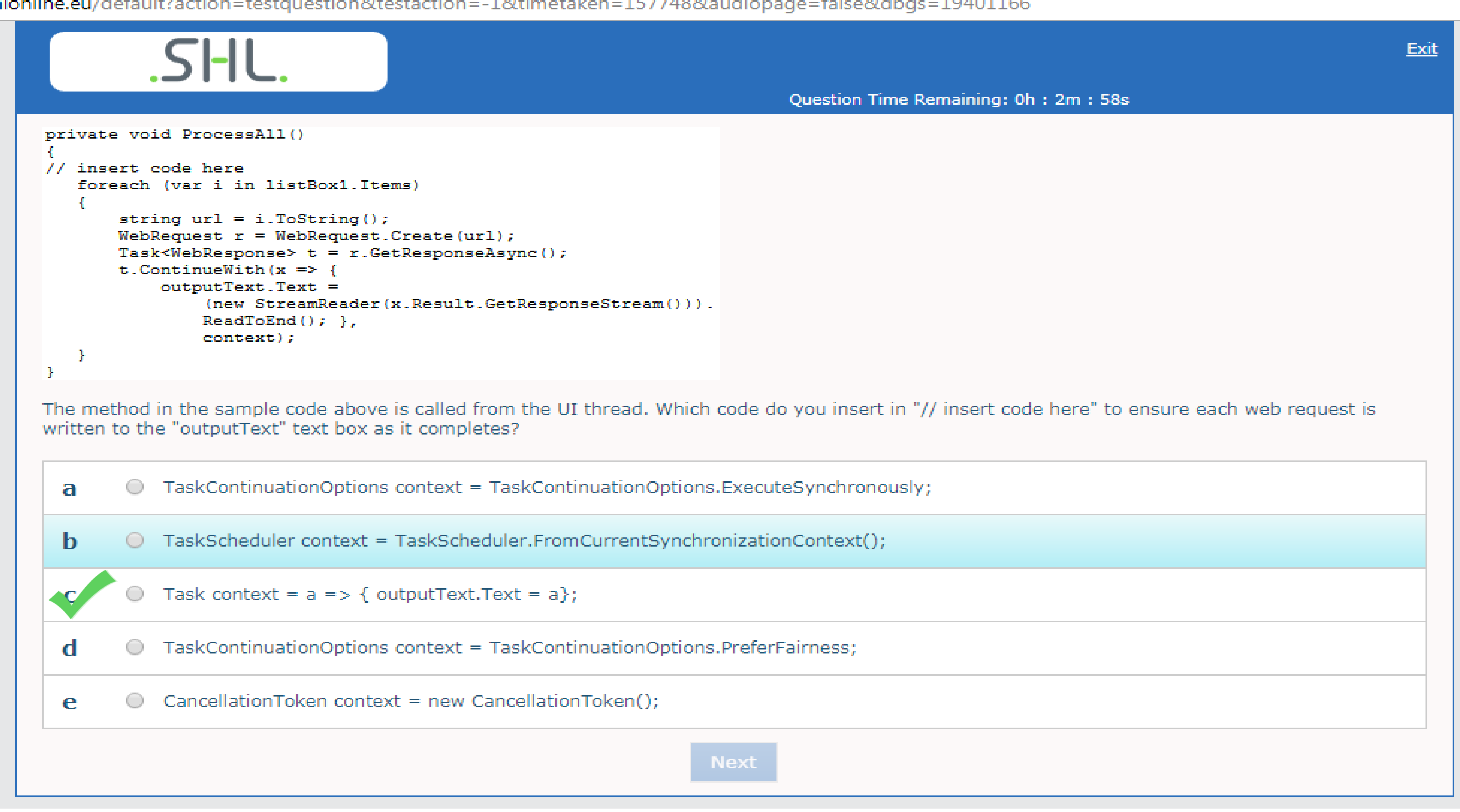
e
```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```



```
static object StartList(object[] array)
   {
       foreach (var o in array)
       {
           if (o is IEnumerable)
           {
               return ((IEnumerable)o).GetEnumerator();
           }
       }
       return Enumerable.Range(0, 4).GetEnumerator();
   }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	○	IEnumerable
b	○	IArray
c	○	IList
d ✓	○	IEnumerator
e	○	ICollection


lonline.eu/default?action=testquestion&testaction=-1&timetaken=157748&audiopage=false&dbgs=19401166

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

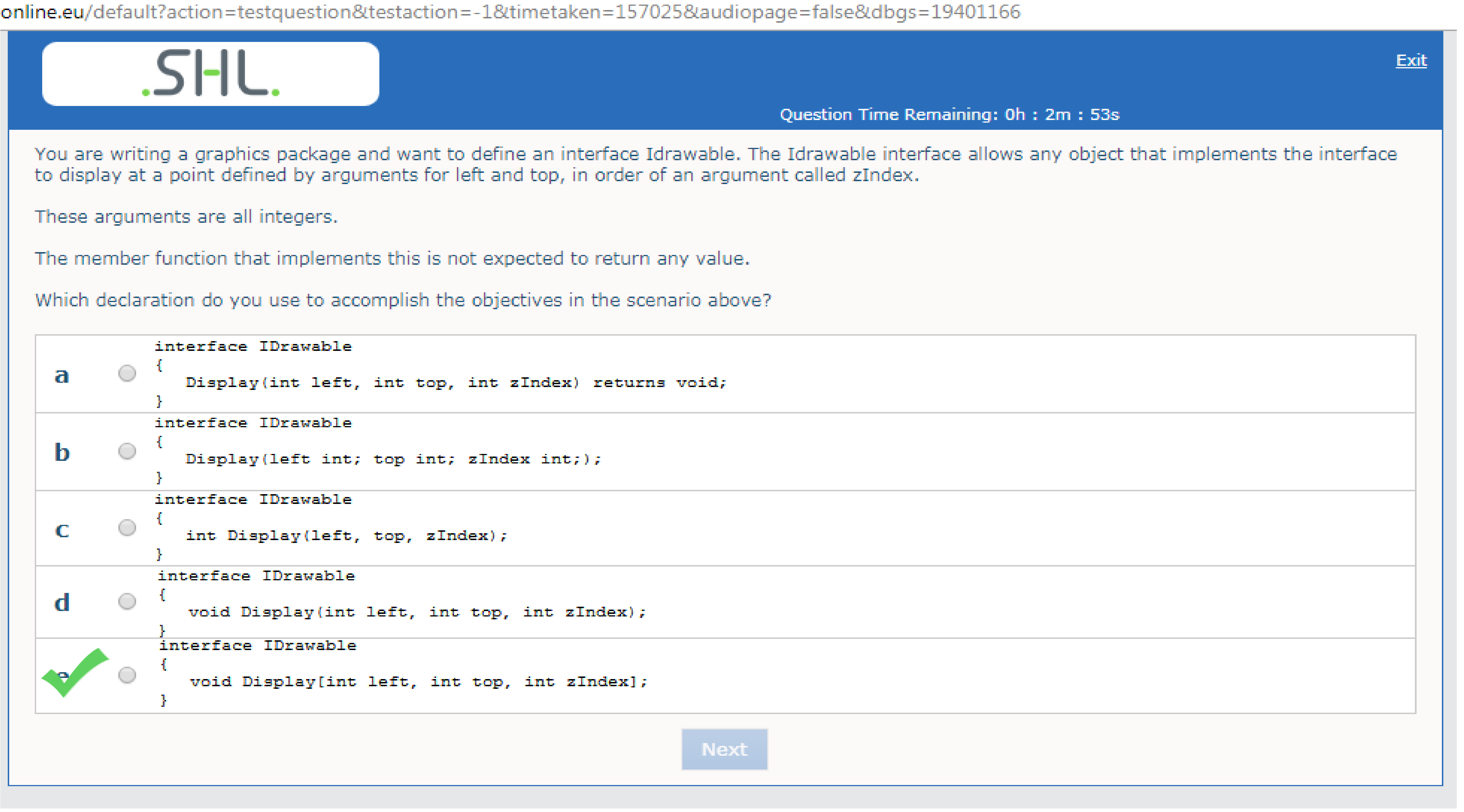
a TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;

b TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();

[✓ c] Task context = a => { outputText.Text = a};

d TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;

e CancellationToken context = new CancellationToken();



online.eu/default?action=testquestion&testaction=-1&timetaken=157025&audiopage=false&dbgs=19401166

You are writing a graphics package and want to define an interface Idrawable. The Idrawable interface allows any object that implements the interface to display at a point defined by arguments for left and top, in order of an argument called zIndex.

These arguments are all integers.

The member function that implements this is not expected to return any value.

Which declaration do you use to accomplish the objectives in the scenario above?

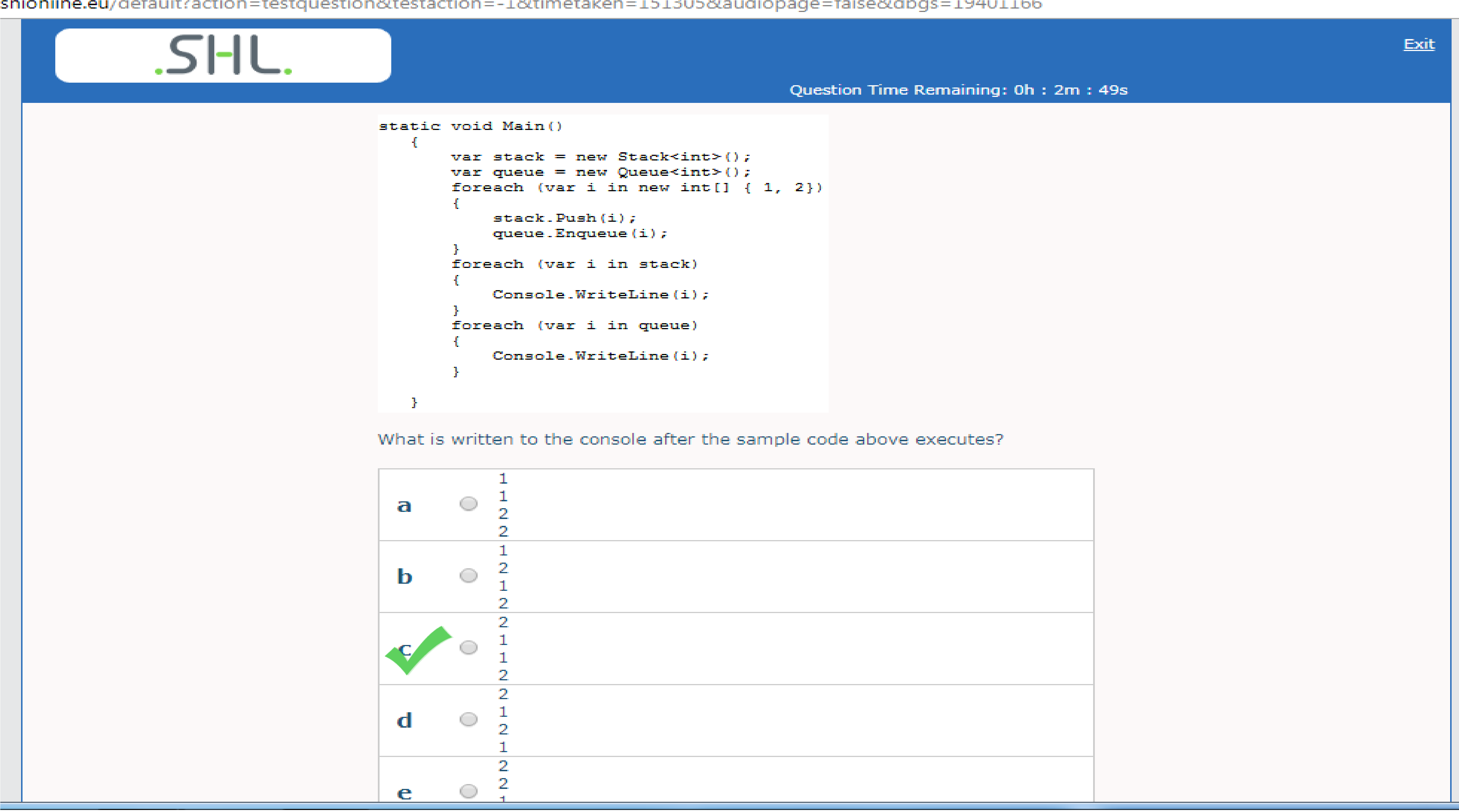
a
```
interface IDrawable
{
   Display(int left, int top, int zIndex) returns void;
}
```

b
```
interface IDrawable
{
   Display(left int; top int; zIndex int;);
}
```

c
```
interface IDrawable
{
   int Display(left, top, zIndex);
}
```

d
```
interface IDrawable
{
   void Display(int left, int top, int zIndex);
}
```

✓ e
```
interface IDrawable
{
   void Display[int left, int top, int zIndex];
}
```



shlonline.eu/default?action=testquestion&testaction=-1&timetaken=151305&audiopage=false&dbgs=19401166

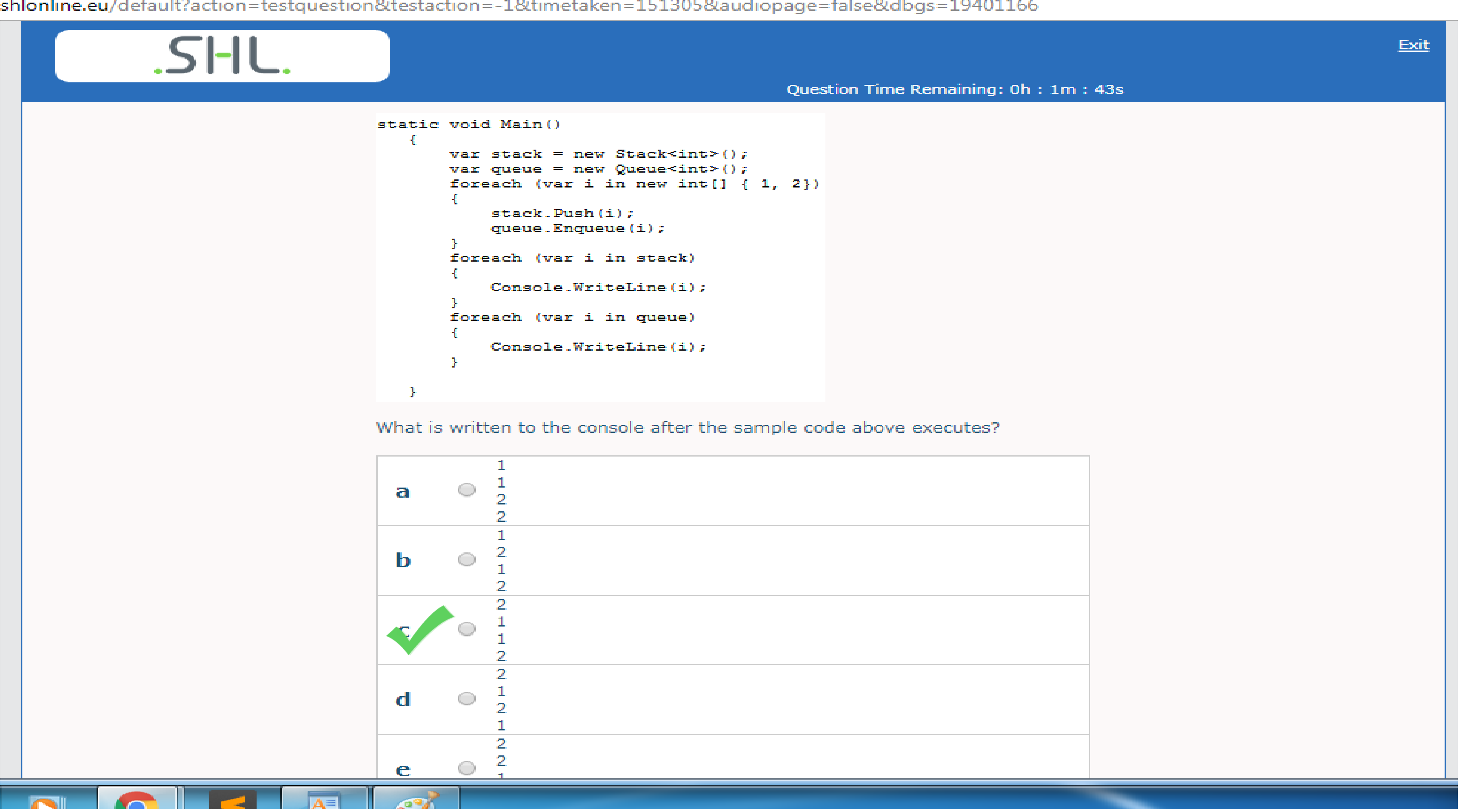
Exit

Question Time Remaining: 0h : 2m : 49s

```
static void Main()
   {
       var stack = new Stack<int>();
       var queue = new Queue<int>();
       foreach (var i in new int[] { 1, 2})
       {
           stack.Push(i);
           queue.Enqueue(i);
       }
       foreach (var i in stack)
       {
           Console.WriteLine(i);
       }
       foreach (var i in queue)
       {
           Console.WriteLine(i);
       }

   }
```

What is written to the console after the sample code above executes?

a	1 1 2 2
b	1 2 1 2
[✓ c]	2 1 1 2
d	2 1 2 1
e	2 2



shlonline.eu/default?action=testquestion&testaction=-1&timetaken=151305&audiopage=false&dbgs=19401166

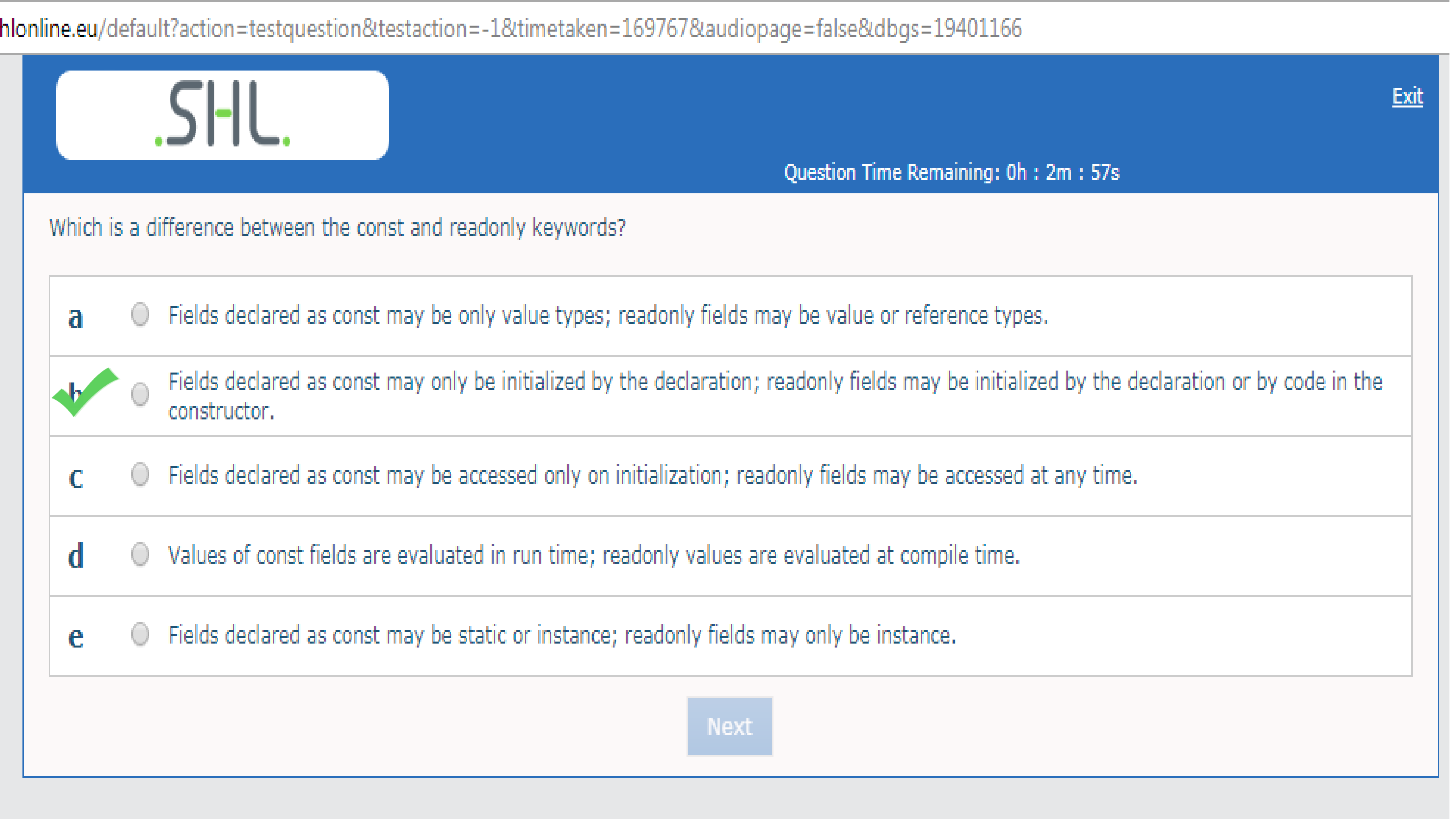
Exit

Question Time Remaining: 0h : 1m : 43s

```
static void Main()
   {
       var stack = new Stack<int>();
       var queue = new Queue<int>();
       foreach (var i in new int[] { 1, 2})
       {
           stack.Push(i);
           queue.Enqueue(i);
       }
       foreach (var i in stack)
       {
           Console.WriteLine(i);
       }
       foreach (var i in queue)
       {
           Console.WriteLine(i);
       }

   }
```

What is written to the console after the sample code above executes?

a	1 1 2 2
b	1 2 1 2
[✓ c]	2 1 1 2
d	2 1 2 1
e	2 2 1


hlonline.eu/default?action=testquestion&testaction=-1&timetaken=169767&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 57s

Which is a difference between the const and readonly keywords?

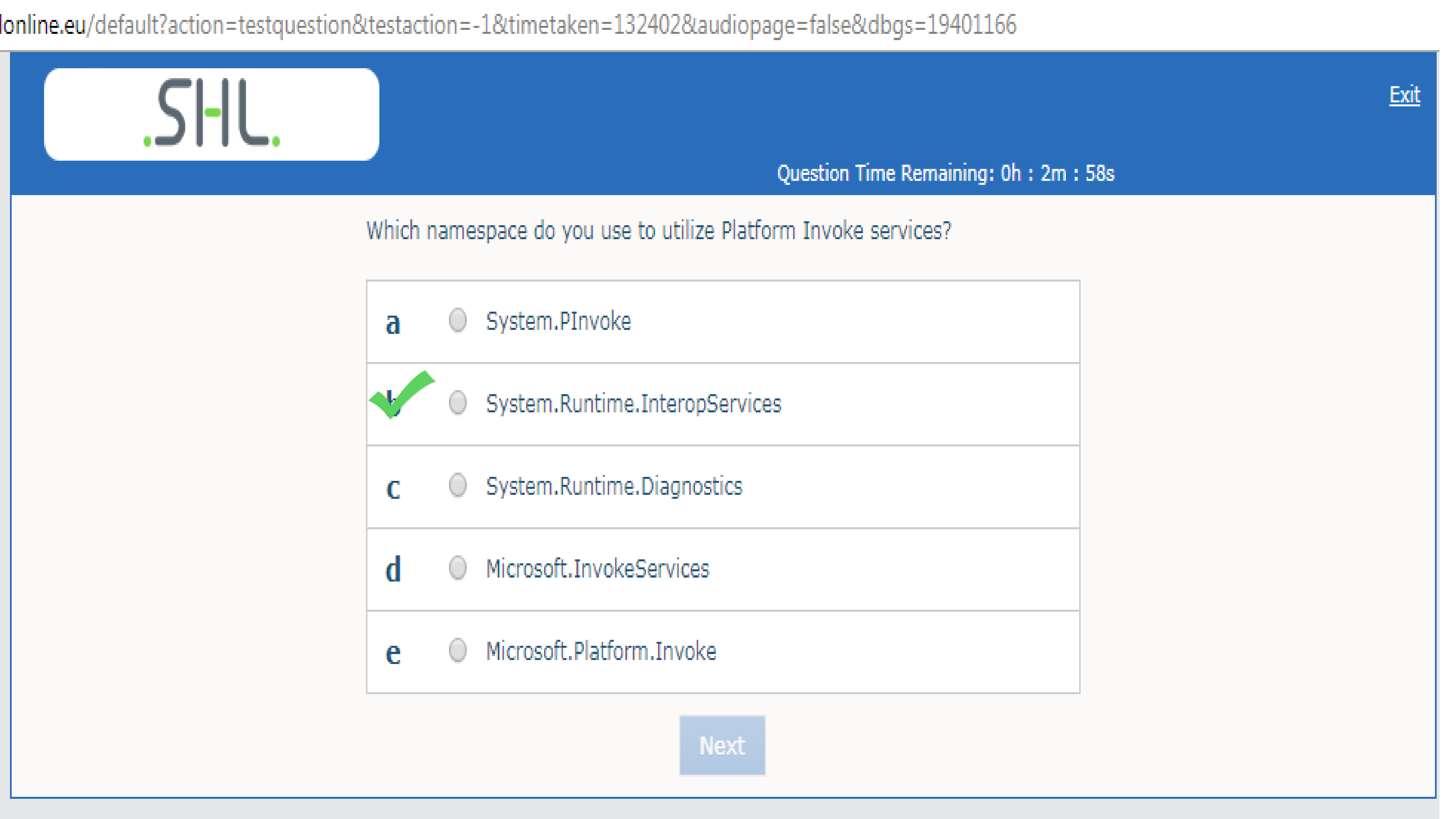
a ○ Fields declared as const may be only value types; readonly fields may be value or reference types.

[✓ b] ○ Fields declared as const may only be initialized by the declaration; readonly fields may be initialized by the declaration or by code in the constructor.

c ○ Fields declared as const may be accessed only on initialization; readonly fields may be accessed at any time.

d ○ Values of const fields are evaluated in run time; readonly values are evaluated at compile time.

e ○ Fields declared as const may be static or instance; readonly fields may only be instance.

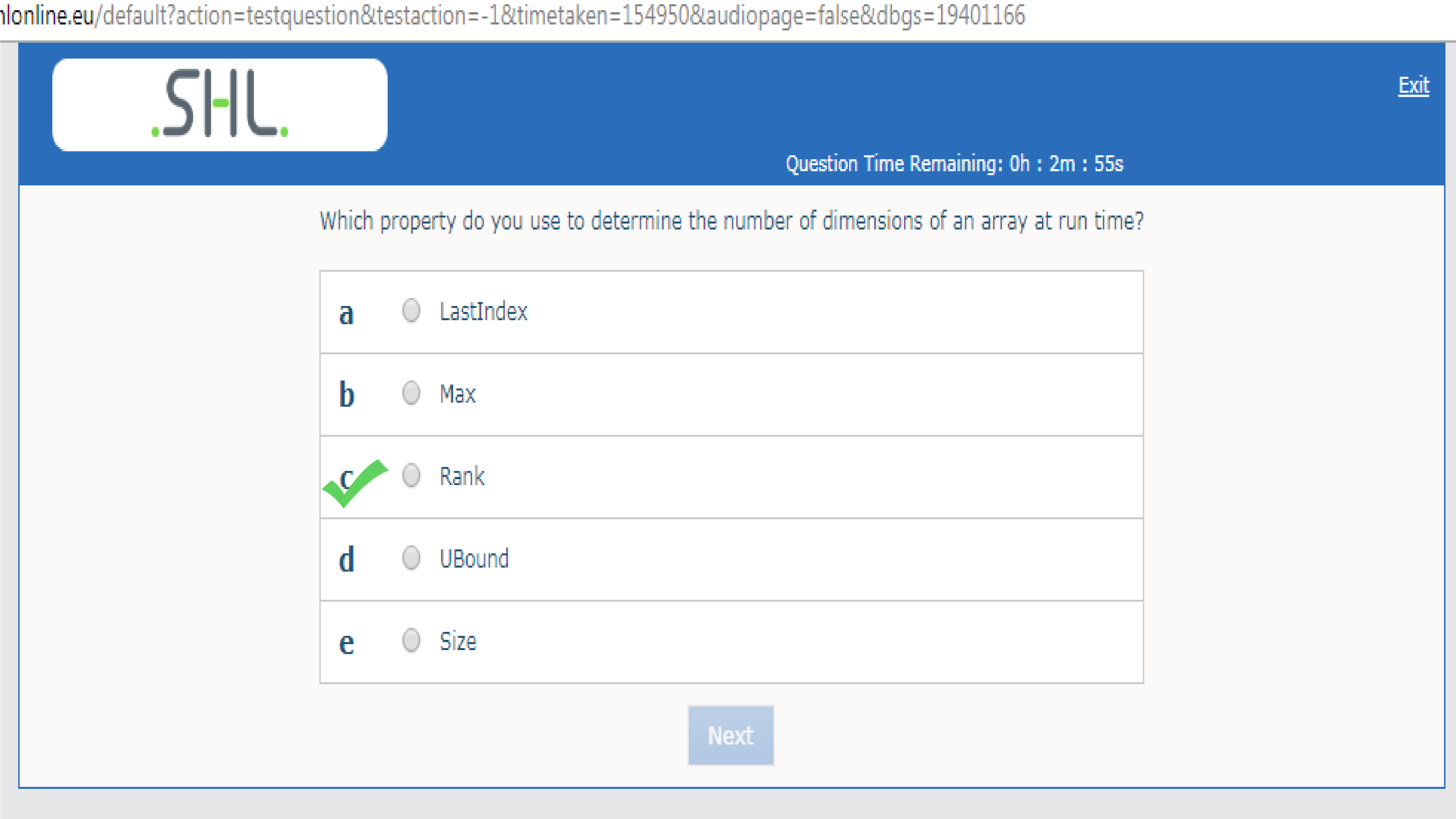


lonline.eu/default?action=testquestion&testaction=-1&timetaken=132402&audiopage=false&dbgs=19401166

Question Time Remaining: 0h : 2m : 58s

Which namespace do you use to utilize Platform Invoke services?

a	System.PInvoke
✓ b	System.Runtime.InteropServices
c	System.Runtime.Diagnostics
d	Microsoft.InvokeServices
e	Microsoft.Platform.Invoke


lonline.eu/default?action=testquestion&testaction=-1&timetaken=154950&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 55s

Which property do you use to determine the number of dimensions of an array at run time?

a LastIndex

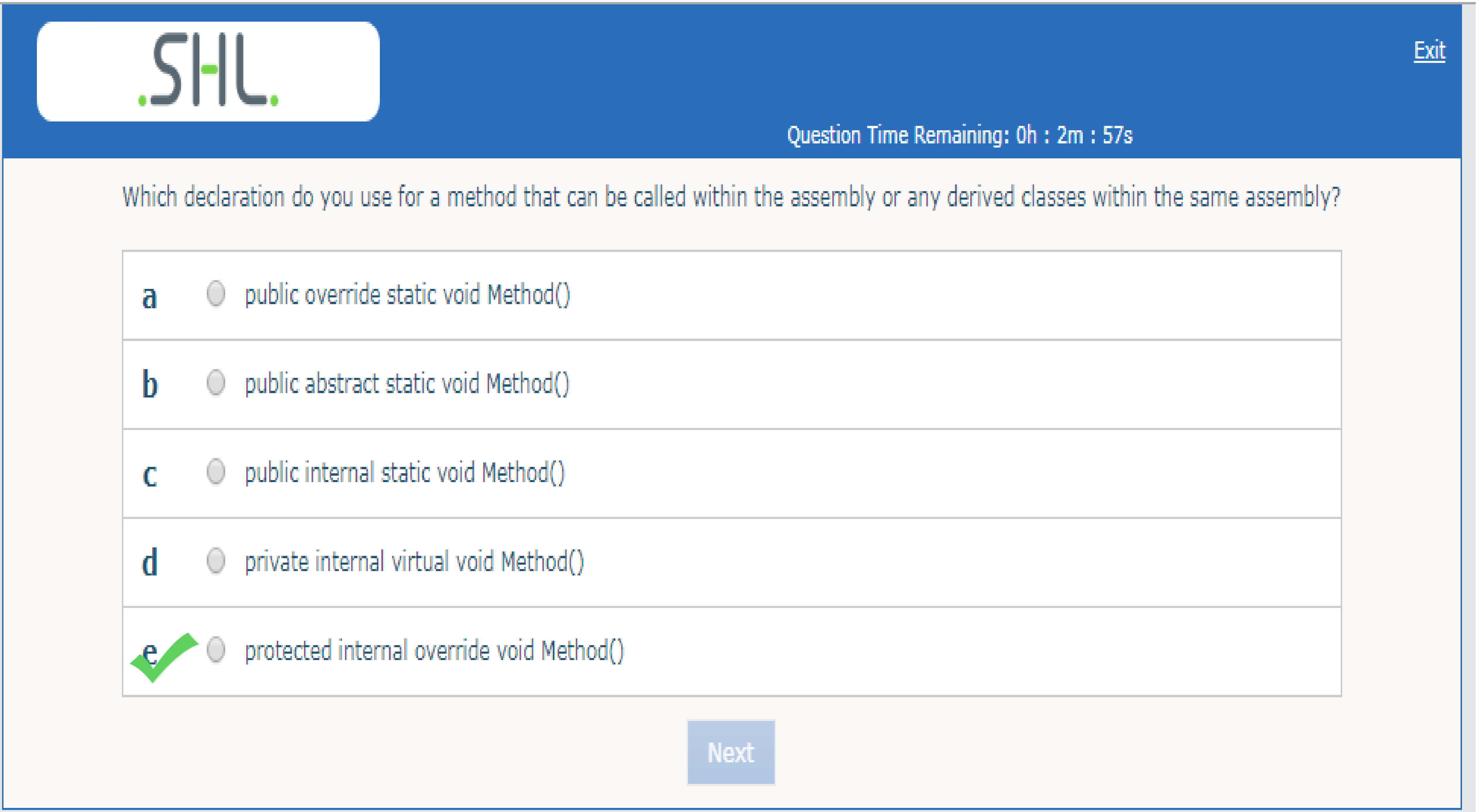
b Max

[✓ c Rank]

d UBound

e Size

Next



Question Time Remaining: 0h : 2m : 57s

Which declaration do you use for a method that can be called within the assembly or any derived classes within the same assembly?

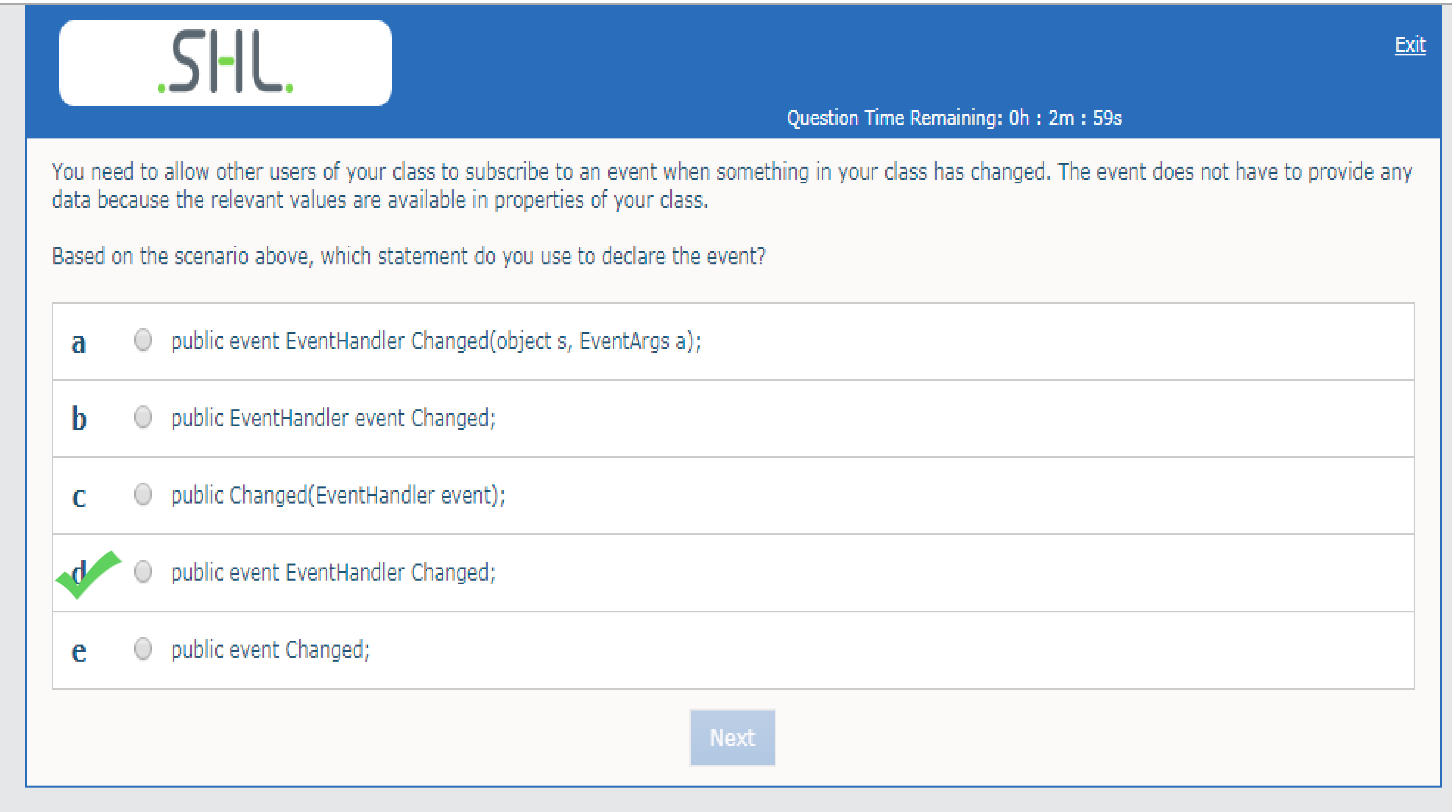
a public override static void Method()

b public abstract static void Method()

c public internal static void Method()

d private internal virtual void Method()

[✓ e] protected internal override void Method()



SHL

Exit

Question Time Remaining: 0h : 2m : 59s

You need to allow other users of your class to subscribe to an event when something in your class has changed. The event does not have to provide any data because the relevant values are available in properties of your class.

Based on the scenario above, which statement do you use to declare the event?

a ○ `public event EventHandler Changed(object s, EventArgs a);`

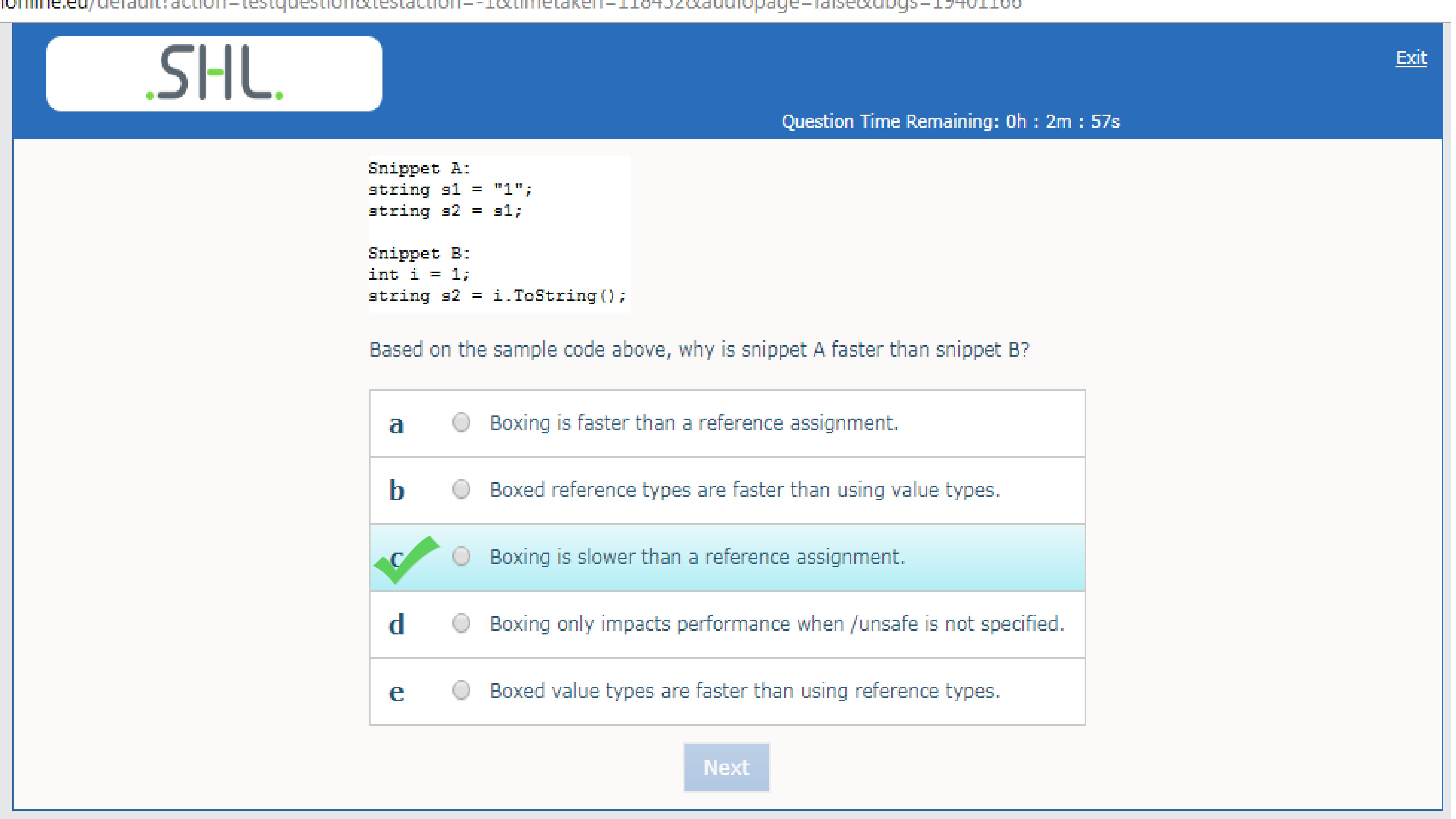
b ○ `public EventHandler event Changed;`

c ○ `public Changed(EventHandler event);`

[✓ d] ○ `public event EventHandler Changed;`

e ○ `public event Changed;`

Next


lonline.eu/default?action=testquestion&testaction=-1&timetaken=118432&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 57s

```
Snippet A:
string s1 = "1";
string s2 = s1;

Snippet B:
int i = 1;
string s2 = i.ToString();
```

Based on the sample code above, why is snippet A faster than snippet B?

a Boxing is faster than a reference assignment.

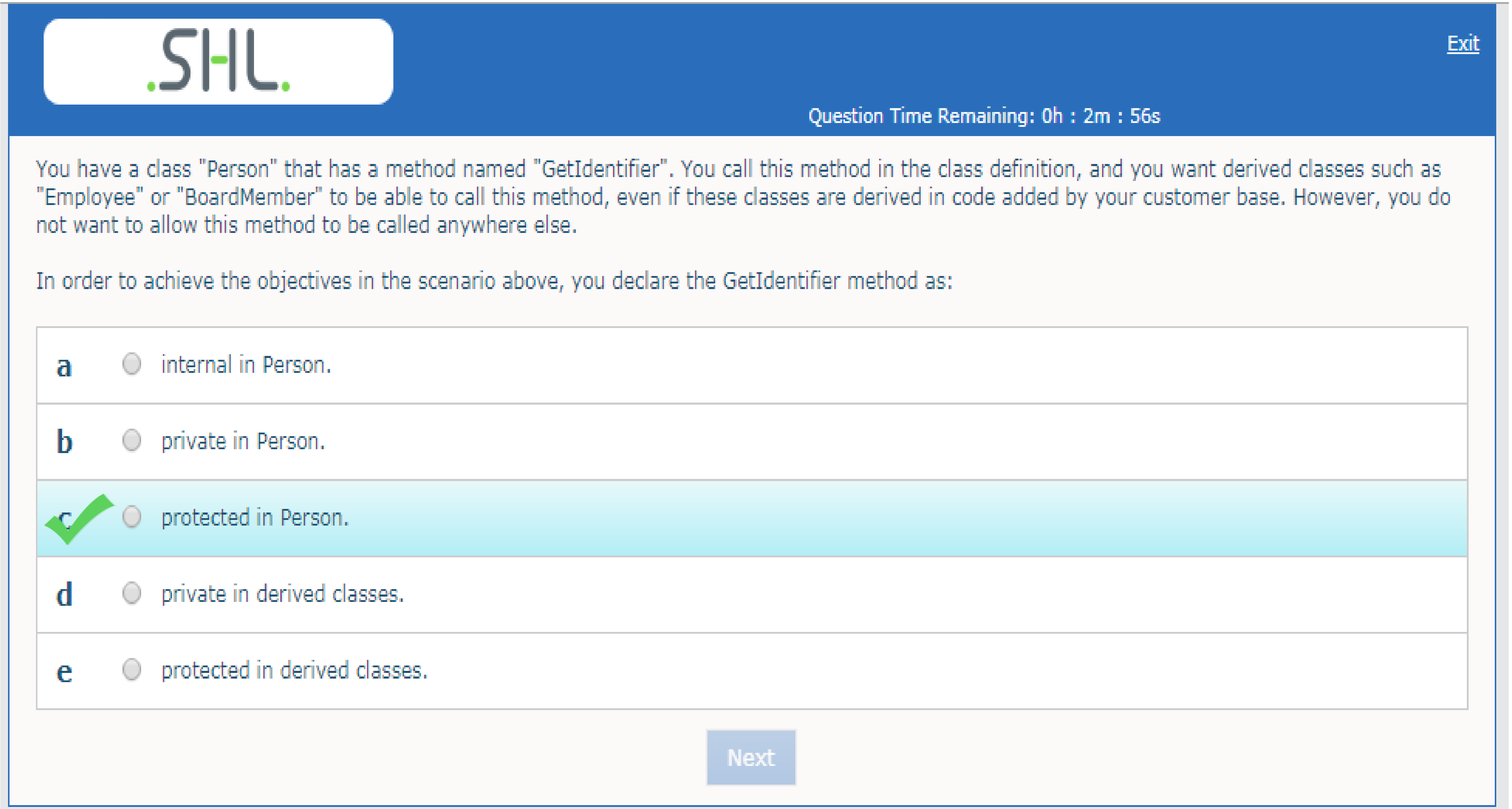
b Boxed reference types are faster than using value types.

[✓ c] Boxing is slower than a reference assignment.

d Boxing only impacts performance when /unsafe is not specified.

e Boxed value types are faster than using reference types.

Next



You have a class "Person" that has a method named "GetIdentifier". You call this method in the class definition, and you want derived classes such as "Employee" or "BoardMember" to be able to call this method, even if these classes are derived in code added by your customer base. However, you do not want to allow this method to be called anywhere else.

In order to achieve the objectives in the scenario above, you declare the GetIdentifier method as:

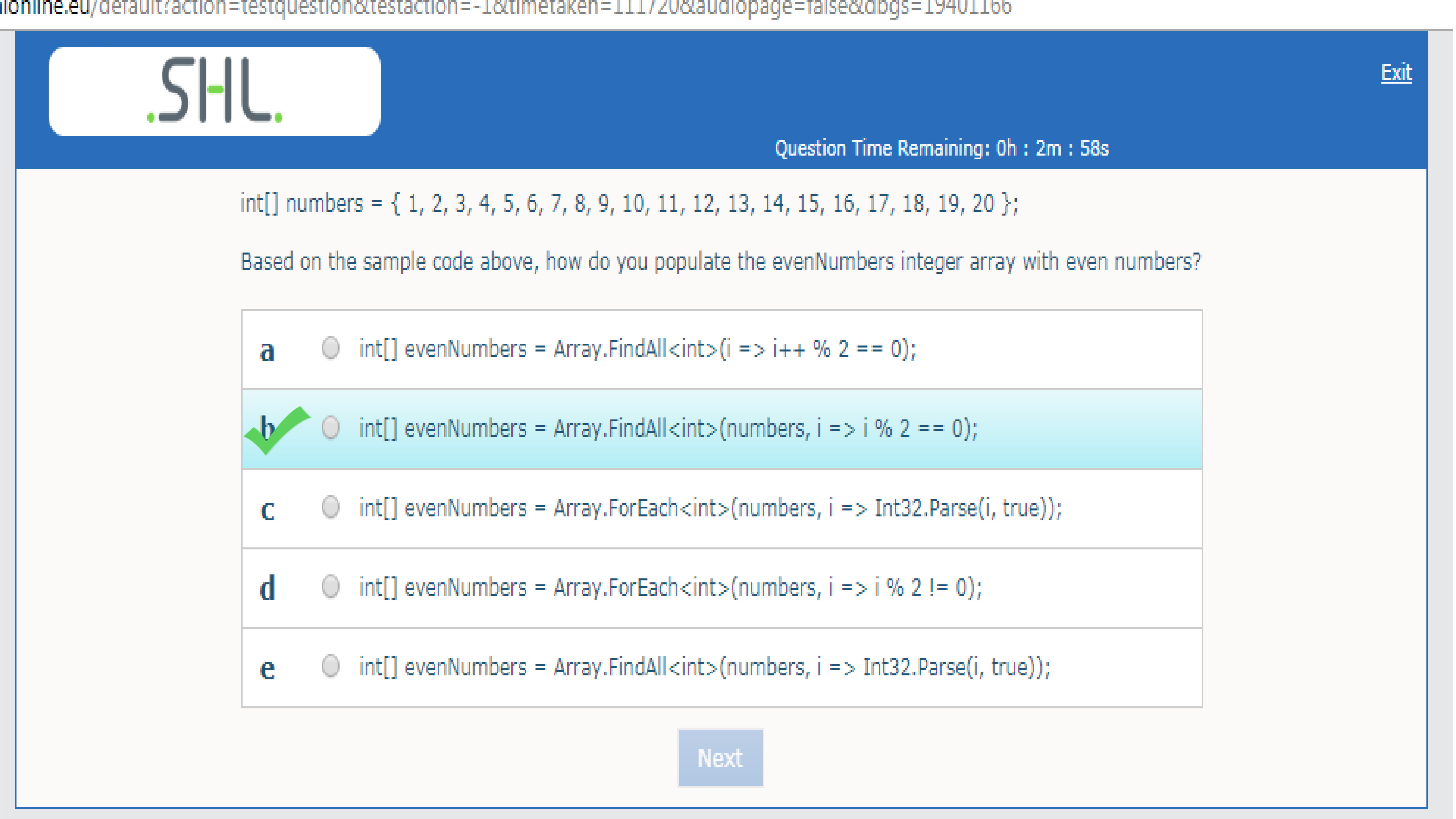
a internal in Person.

b private in Person.

[✓ c protected in Person.]

d private in derived classes.

e protected in derived classes.



ionline.eu/default?action=testquestion&testaction=-1&timetaken=111720&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 58s

```
int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };
```

Based on the sample code above, how do you populate the evenNumbers integer array with even numbers?

a `int[] evenNumbers = Array.FindAll<int>(i => i++ % 2 == 0);`

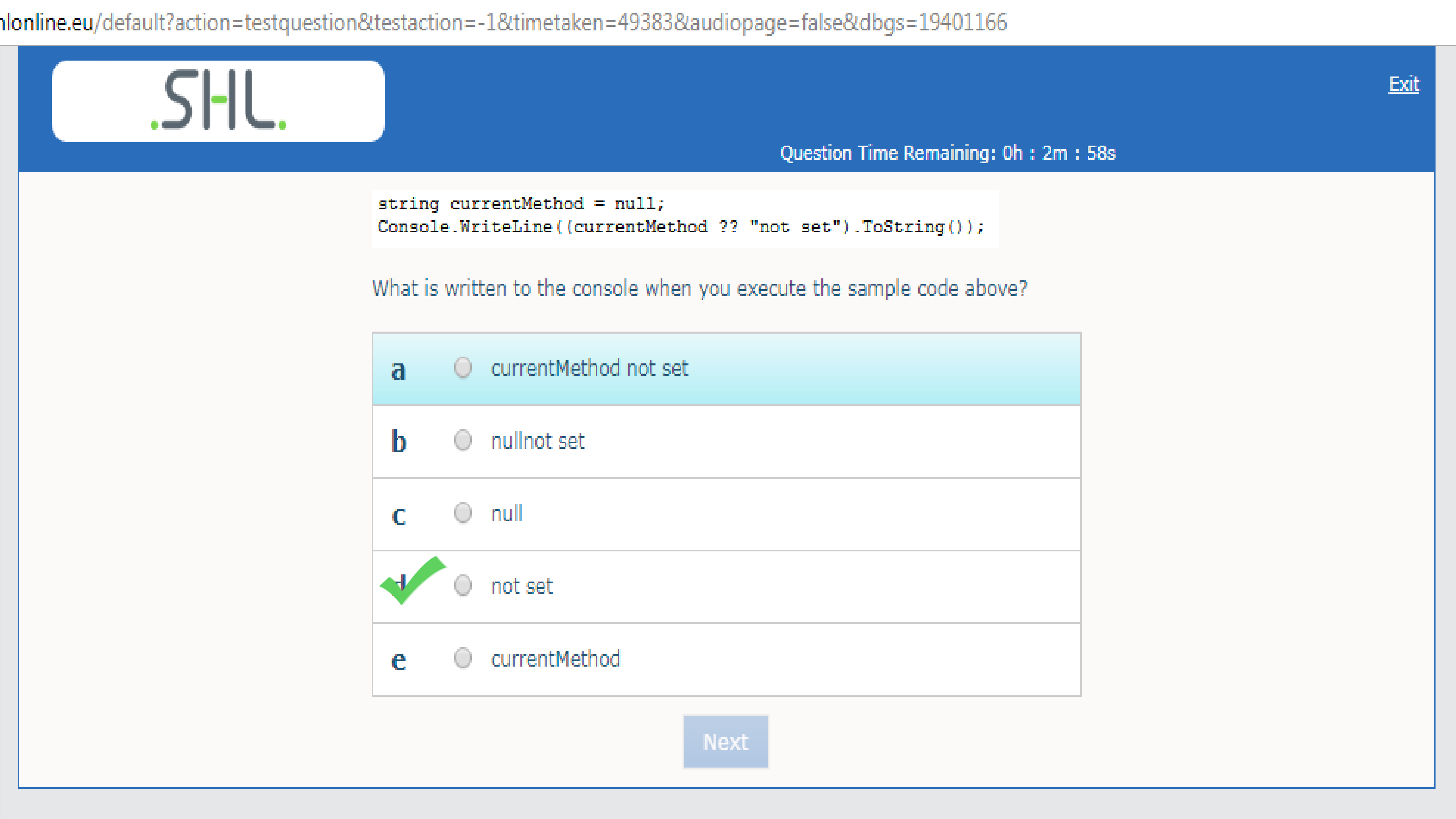
b ✓ `int[] evenNumbers = Array.FindAll<int>(numbers, i => i % 2 == 0);`

c `int[] evenNumbers = Array.ForEach<int>(numbers, i => Int32.Parse(i, true));`

d `int[] evenNumbers = Array.ForEach<int>(numbers, i => i % 2 != 0);`

e `int[] evenNumbers = Array.FindAll<int>(numbers, i => Int32.Parse(i, true));`

Next



nlonline.eu/default?action=testquestion&testaction=-1&timetaken=49383&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 58s

```
string currentMethod = null;
Console.WriteLine((currentMethod ?? "not set").ToString());
```

What is written to the console when you execute the sample code above?

a currentMethod not set

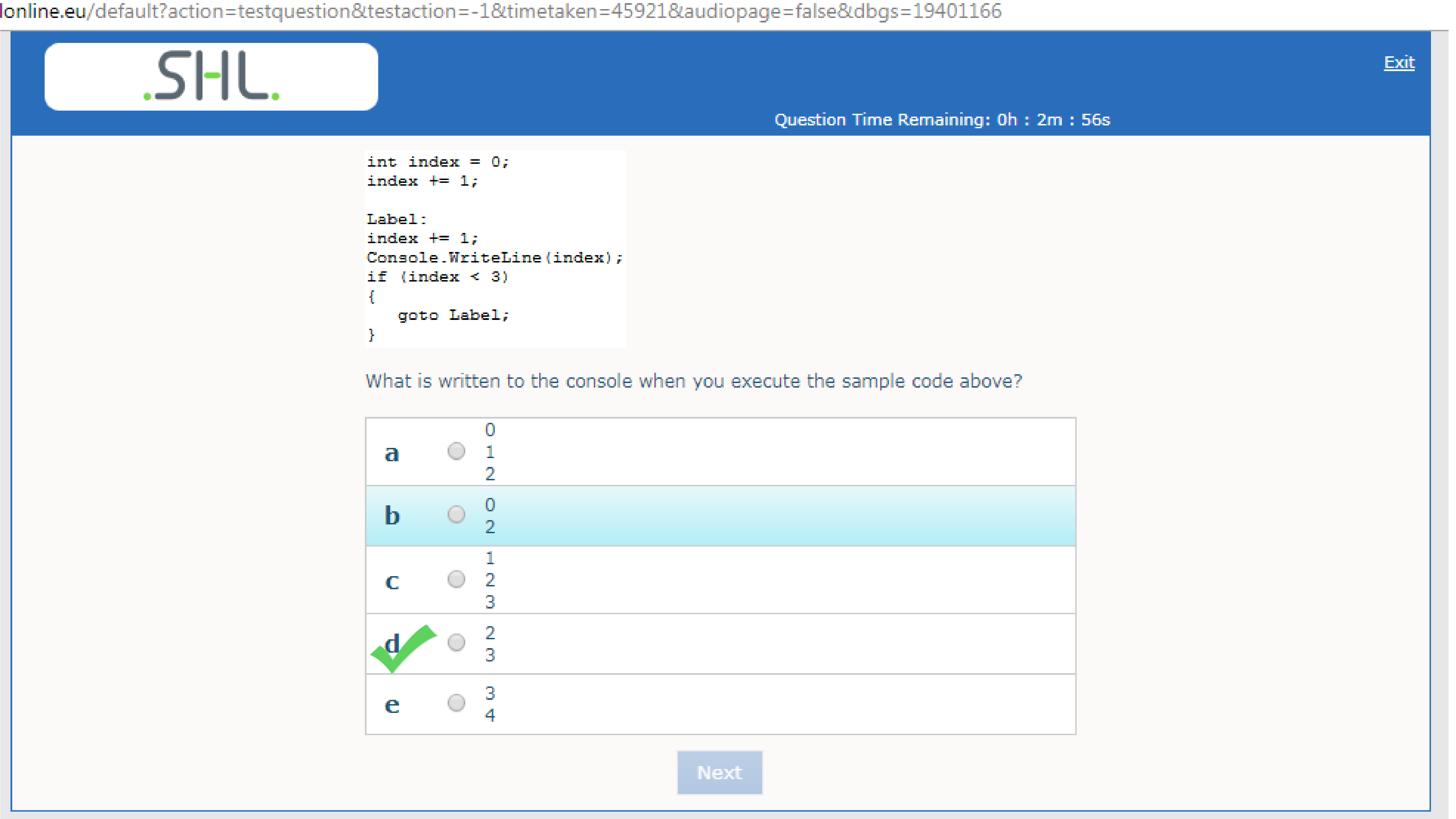
b nullnot set

c null

[✓ d not set]

e currentMethod

Next

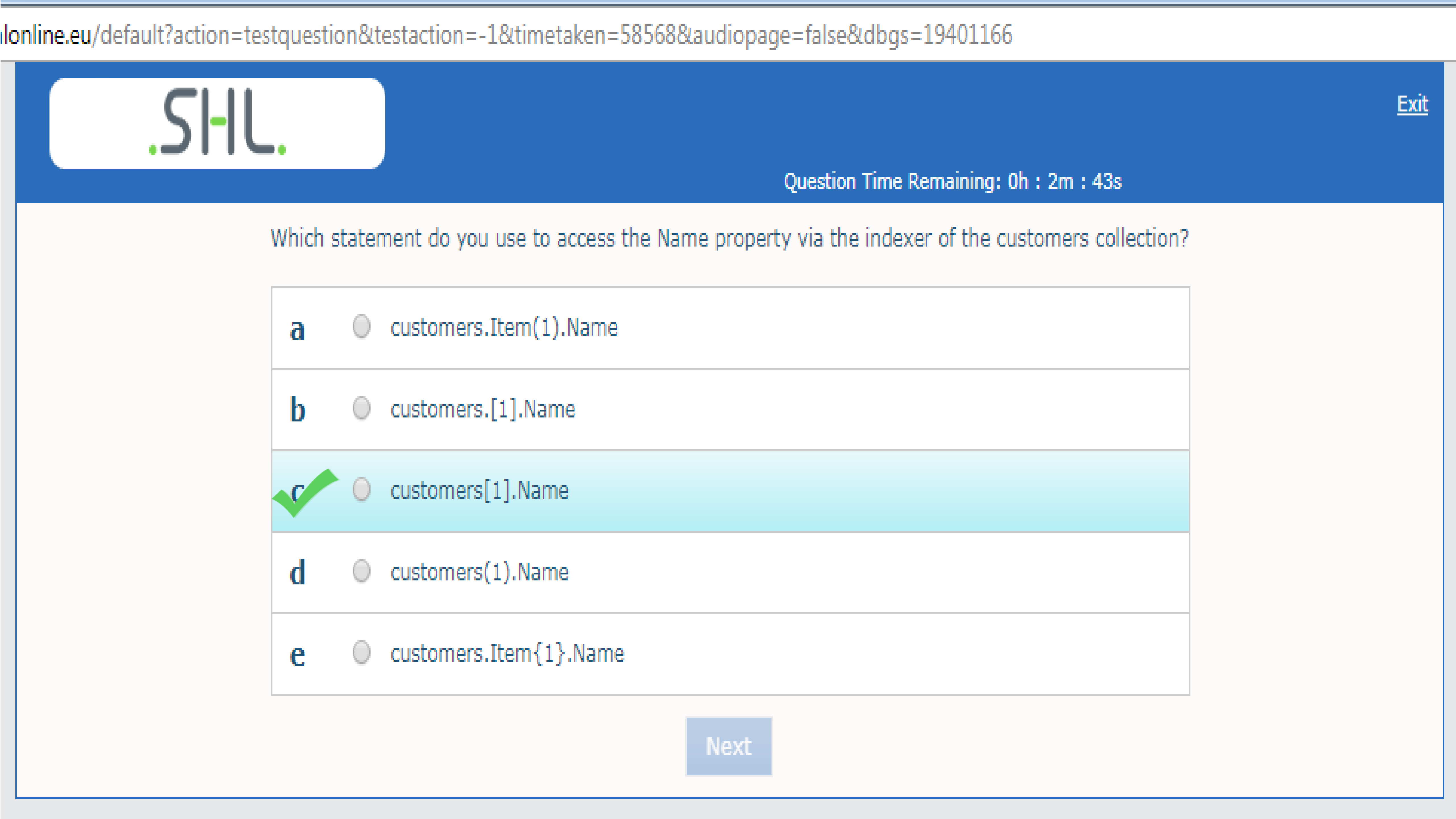

lonline.eu/default?action=testquestion&testaction=-1&timetaken=45921&audiopage=false&dbgs=19401166
Exit
Next

Question Time Remaining: 0h : 2m : 56s

```
int index = 0;
index += 1;

Label:
index += 1;
Console.WriteLine(index);
if (index < 3)
{
   goto Label;
}
```

What is written to the console when you execute the sample code above?

a	0 1 2
b	0 2
c	1 2 3
[✓ d]	2 3
e	3 4



lonline.eu/default?action=testquestion&testaction=-1&timetaken=58568&audiopage=false&dbgs=19401166

Question Time Remaining: 0h : 2m : 43s

Which statement do you use to access the Name property via the indexer of the customers collection?

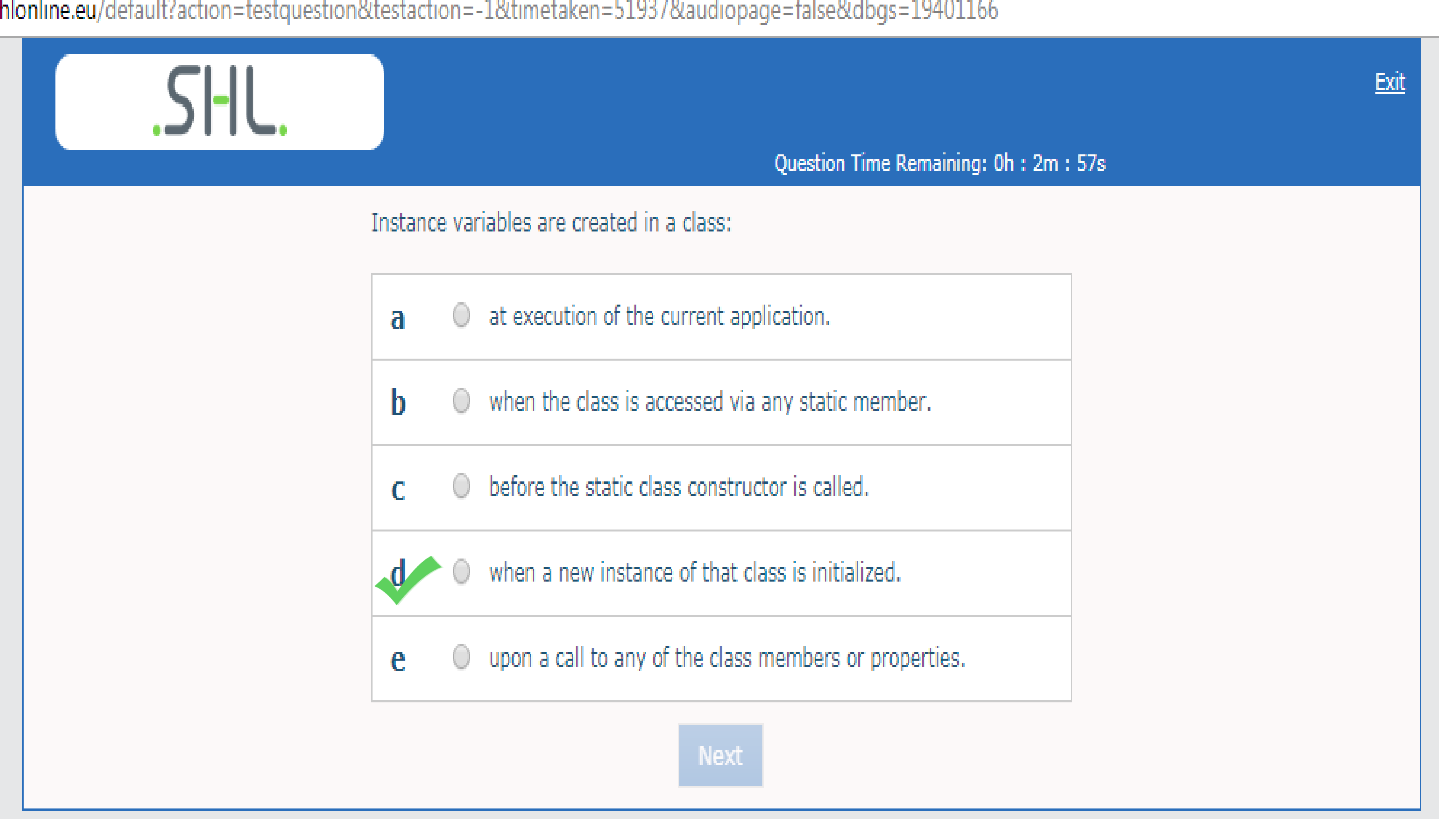
a customers.Item(1).Name

b customers.[1].Name

[✓ c customers[1].Name]

d customers(1).Name

e customers.Item{1}.Name



hlonline.eu/default?action=testquestion&testaction=-1&timetaken=51937&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 57s

Instance variables are created in a class:

a ○ at execution of the current application.

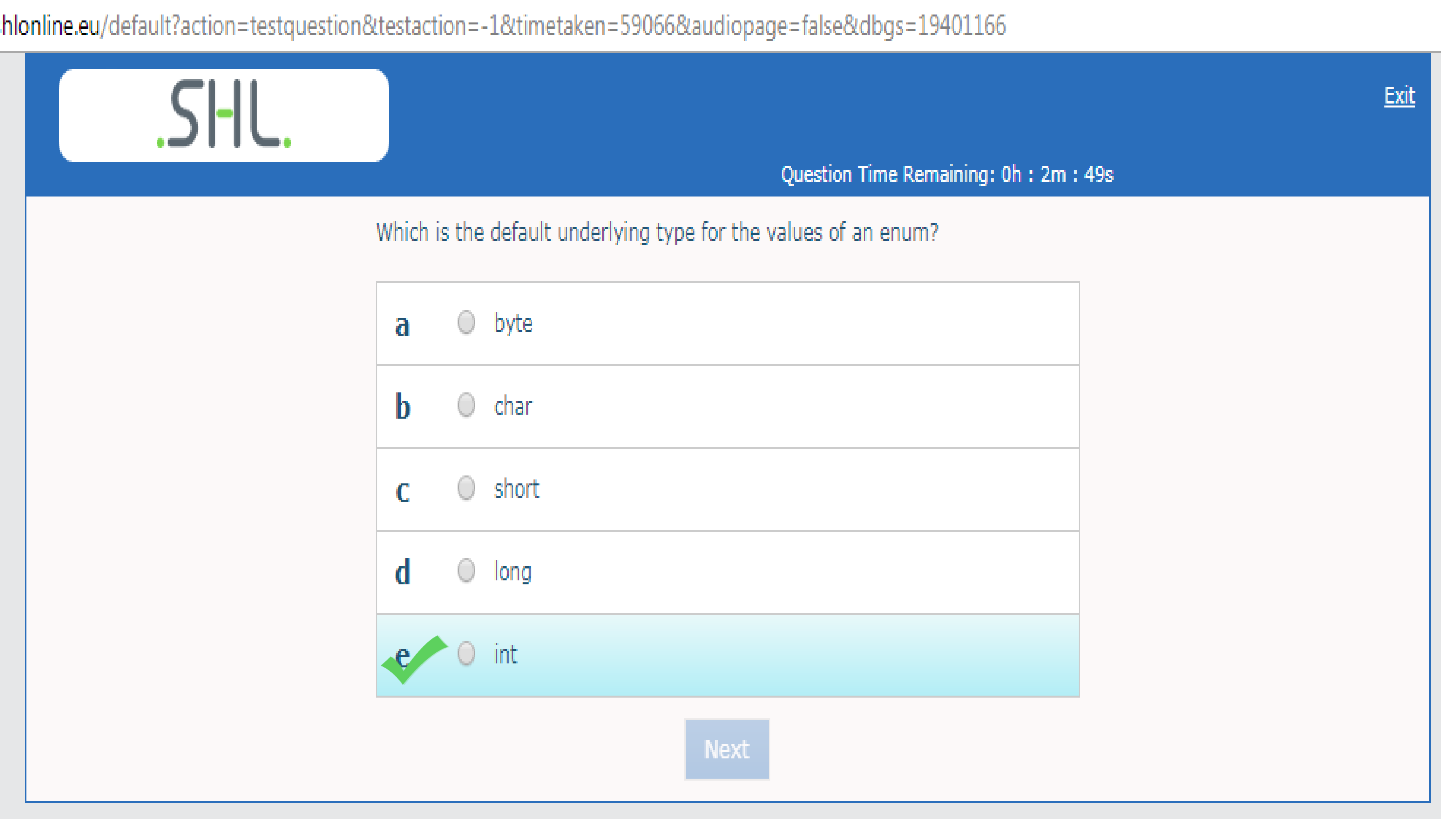
b ○ when the class is accessed via any static member.

c ○ before the static class constructor is called.

d ✓ ○ when a new instance of that class is initialized.

e ○ upon a call to any of the class members or properties.

Next

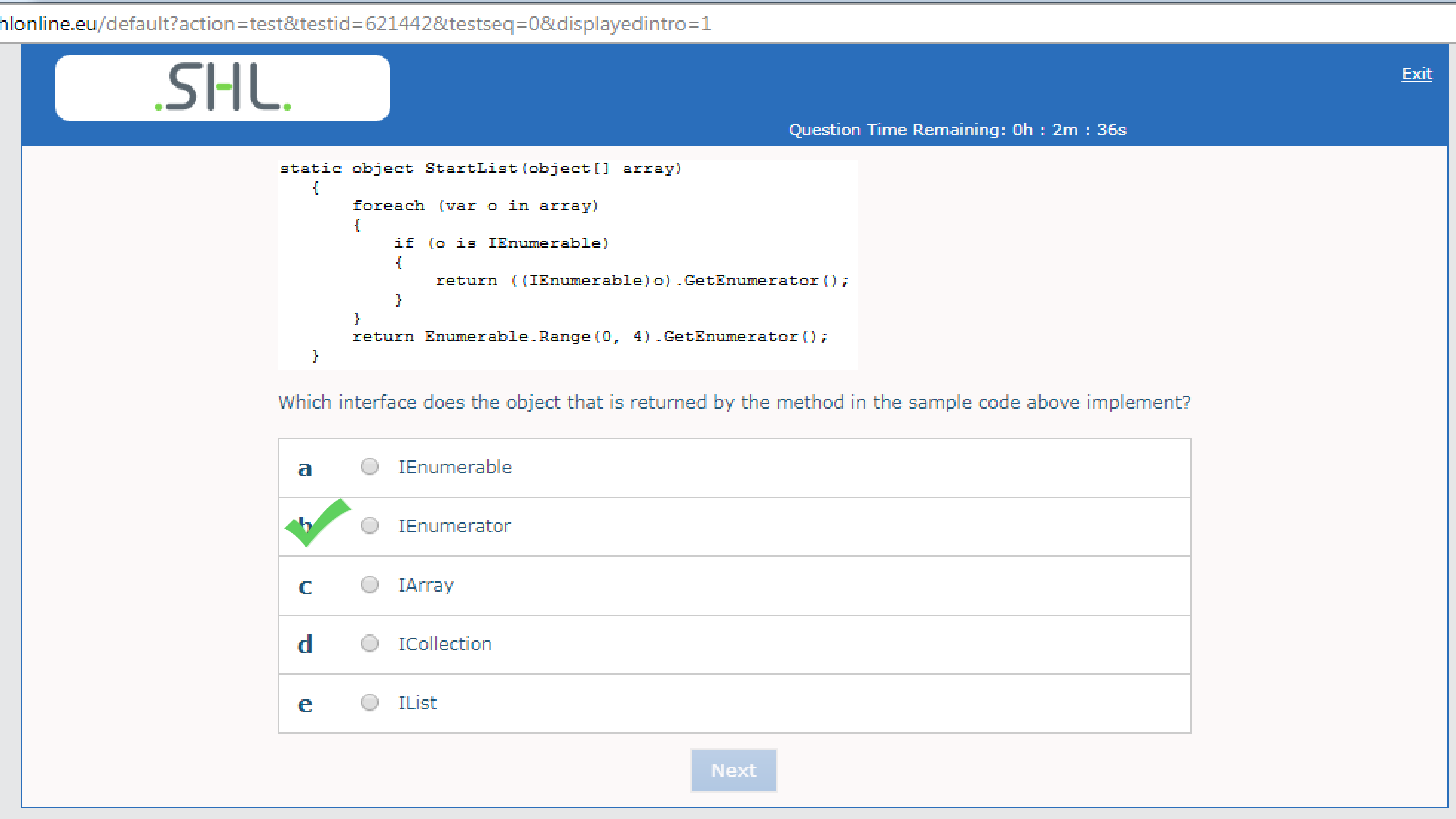

hlonline.eu/default?action=testquestion&testaction=-1&timetaken=59066&audiopage=false&dbgs=19401166

Exit

Question Time Remaining: 0h : 2m : 49s

Which is the default underlying type for the values of an enum?

a	○ byte
b	○ char
c	○ short
d	○ long
e ✓	○ int



hlonline.eu/default?action=test&testid=621442&testseq=0&displayedintro=1

SHL.

Exit

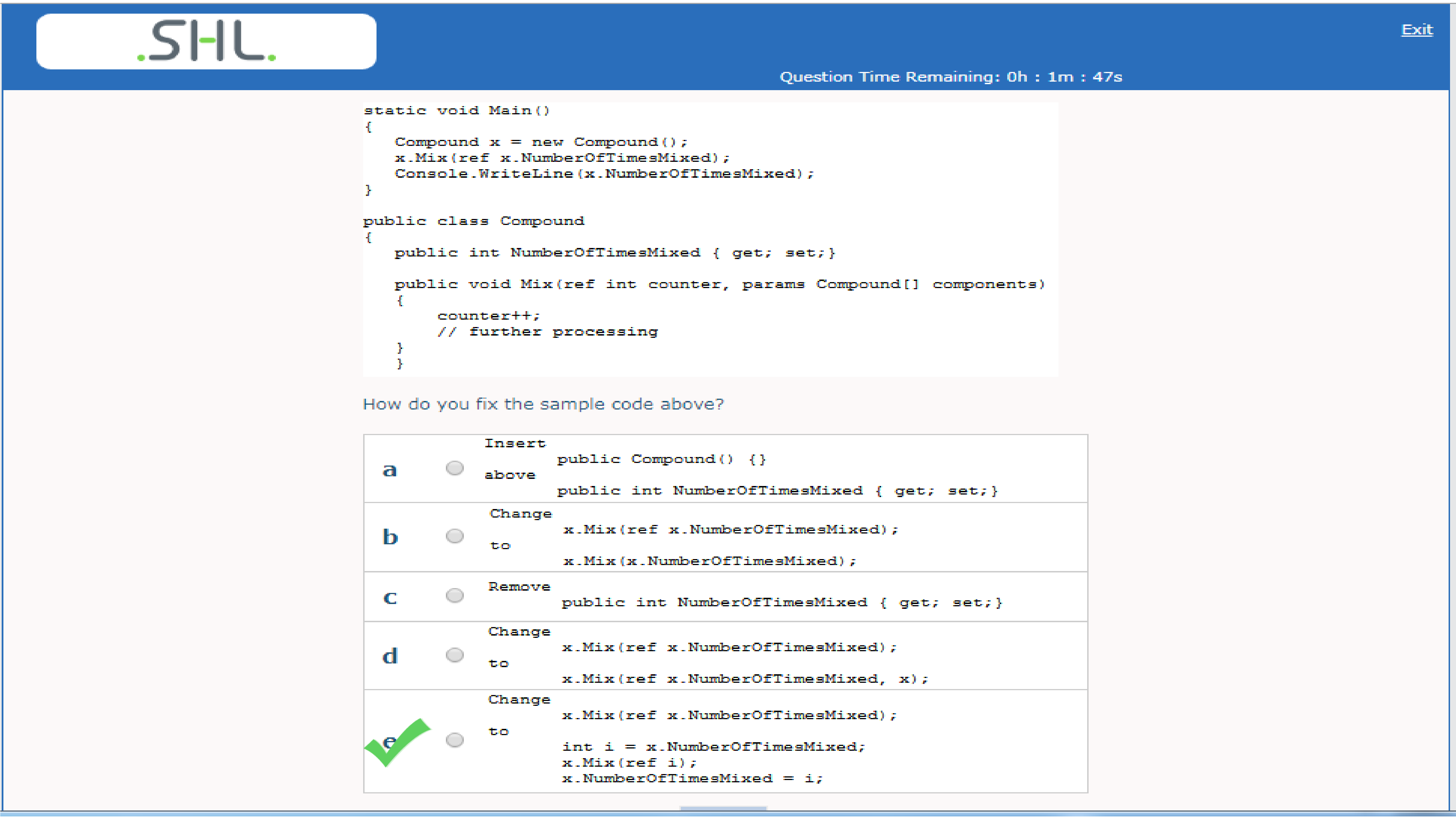
Question Time Remaining: 0h : 2m : 36s

```
static object StartList(object[] array)
   {
       foreach (var o in array)
       {
           if (o is IEnumerable)
           {
               return ((IEnumerable)o).GetEnumerator();
           }
       }
       return Enumerable.Range(0, 4).GetEnumerator();
   }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	IEnumerable
[✓ b]	IEnumerator
c	IArray
d	ICollection
e	IList

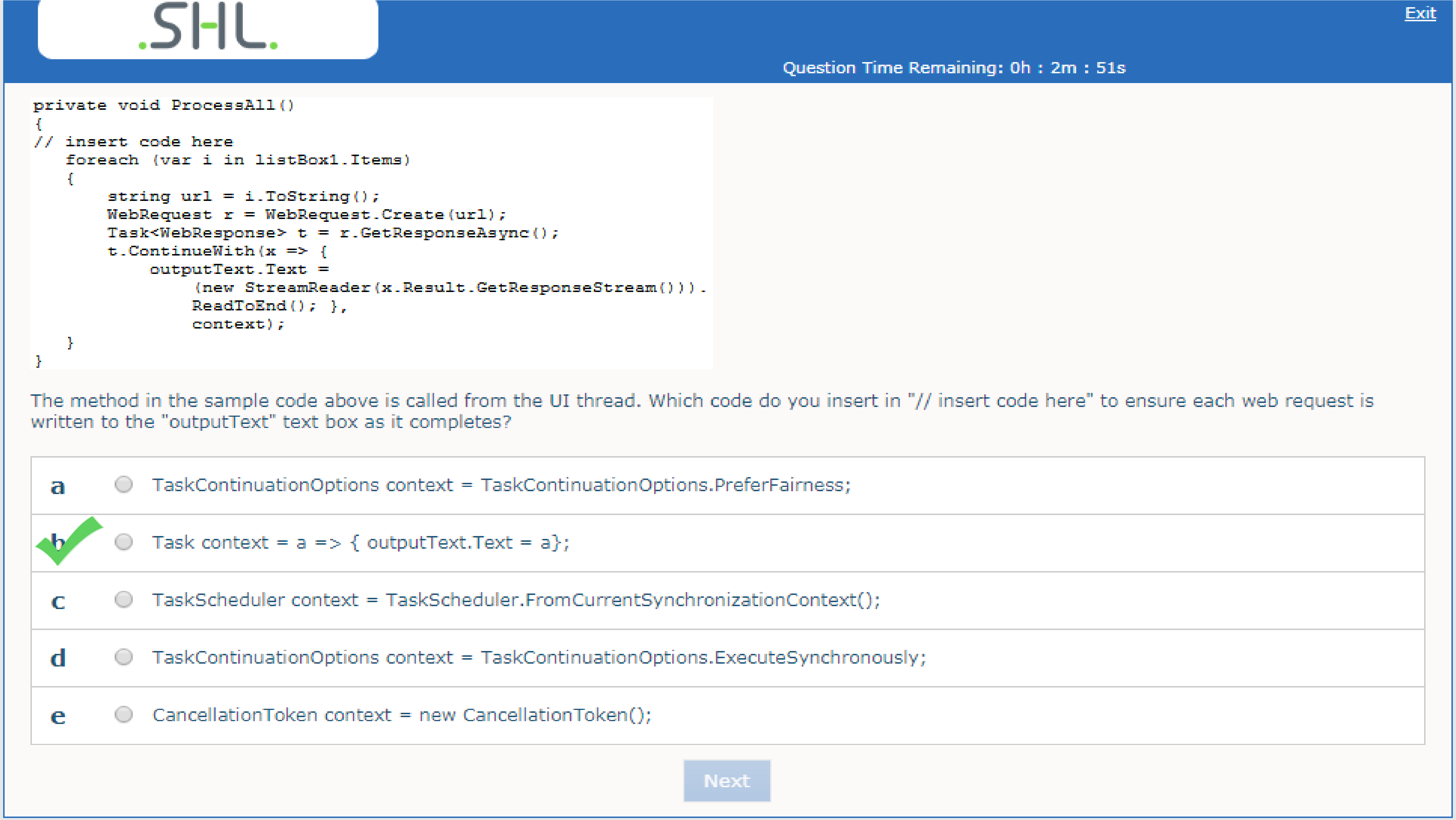
Next



```
static void Main()
{
   Compound x = new Compound();
   x.Mix(ref x.NumberOfTimesMixed);
   Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
   public int NumberOfTimesMixed { get; set;}

   public void Mix(ref int counter, params Compound[] components)
   {
       counter++;
       // further processing
   }
   }
```

How do you fix the sample code above?

a	○	Insert `public Compound() {}` above `public int NumberOfTimesMixed { get; set;}`
b	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(x.NumberOfTimesMixed);`
c	○	Remove `public int NumberOfTimesMixed { get; set;}`
d	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(ref x.NumberOfTimesMixed, x);`
✓ e	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `int i = x.NumberOfTimesMixed; x.Mix(ref i); x.NumberOfTimesMixed = i;`



```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

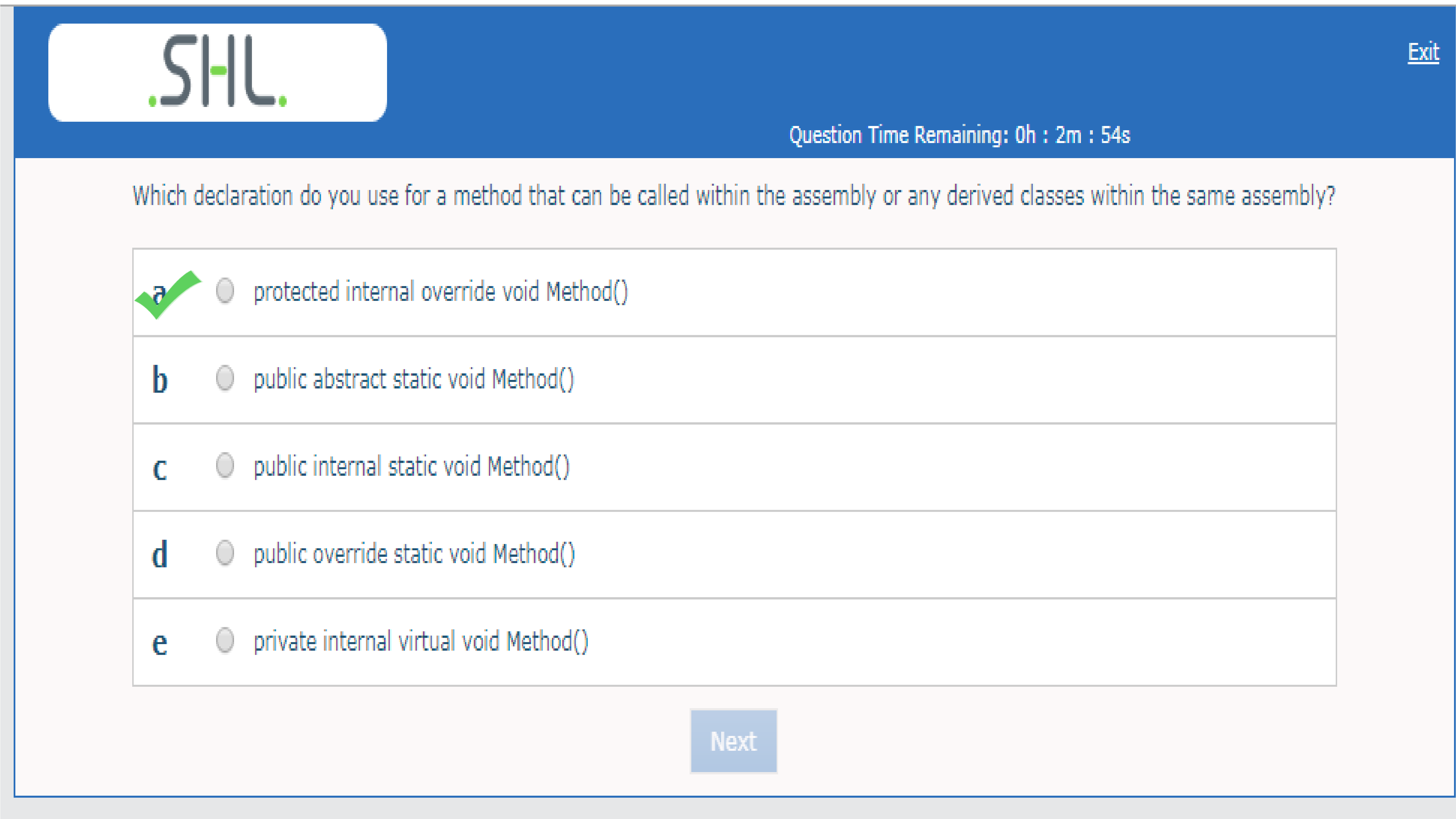
a ○ TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;

b ✓ ○ Task context = a => { outputText.Text = a};

c ○ TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();

d ○ TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;

e ○ CancellationToken context = new CancellationToken();



SHL

Exit

Question Time Remaining: 0h : 2m : 54s

Which declaration do you use for a method that can be called within the assembly or any derived classes within the same assembly?

[✓ a] protected internal override void Method()

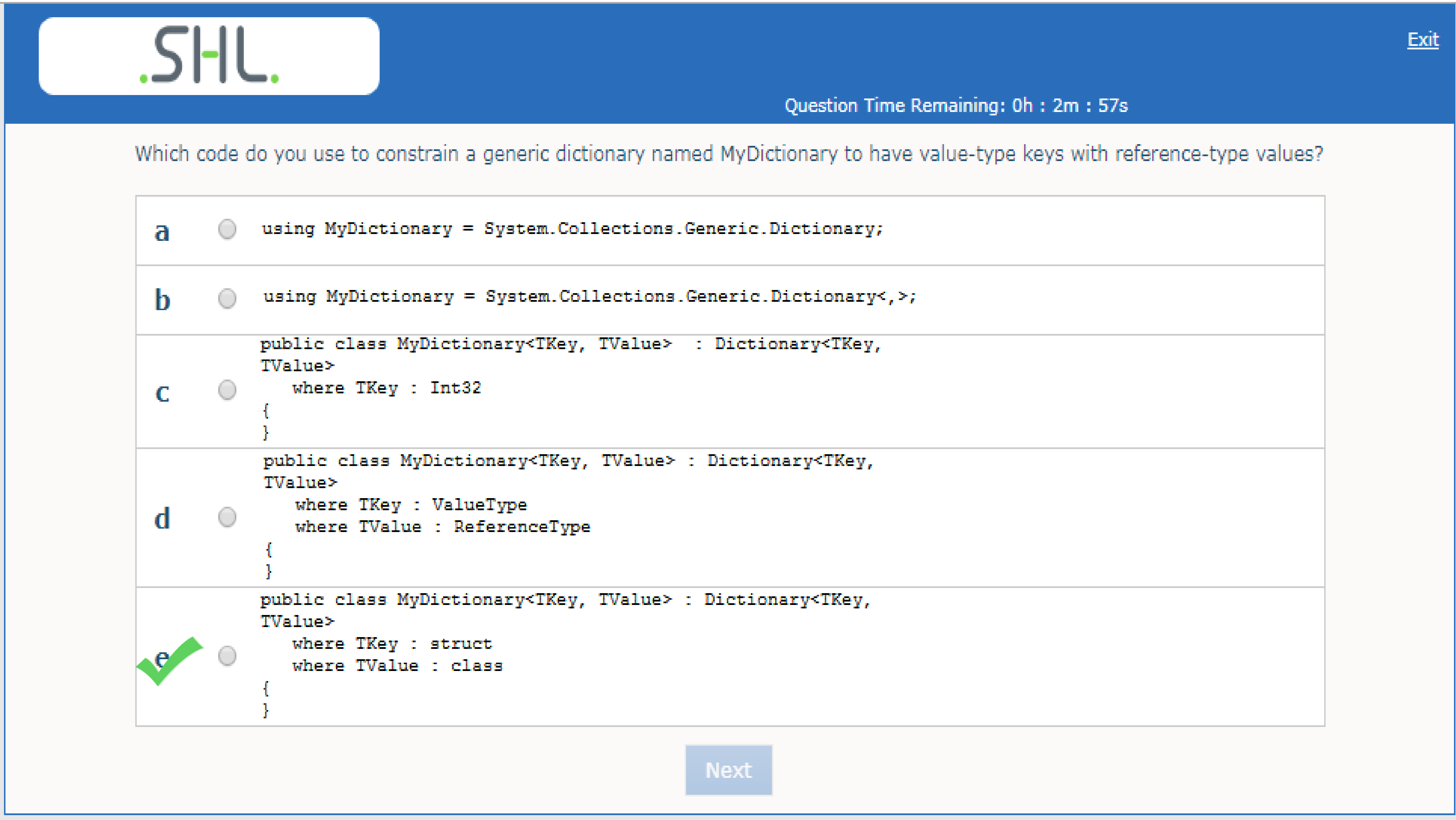
b public abstract static void Method()

c public internal static void Method()

d public override static void Method()

e private internal virtual void Method()

Next



Exit

Question Time Remaining: 0h : 2m : 57s

Which code do you use to constrain a generic dictionary named MyDictionary to have value-type keys with reference-type values?

a
```
using MyDictionary = System.Collections.Generic.Dictionary;
```

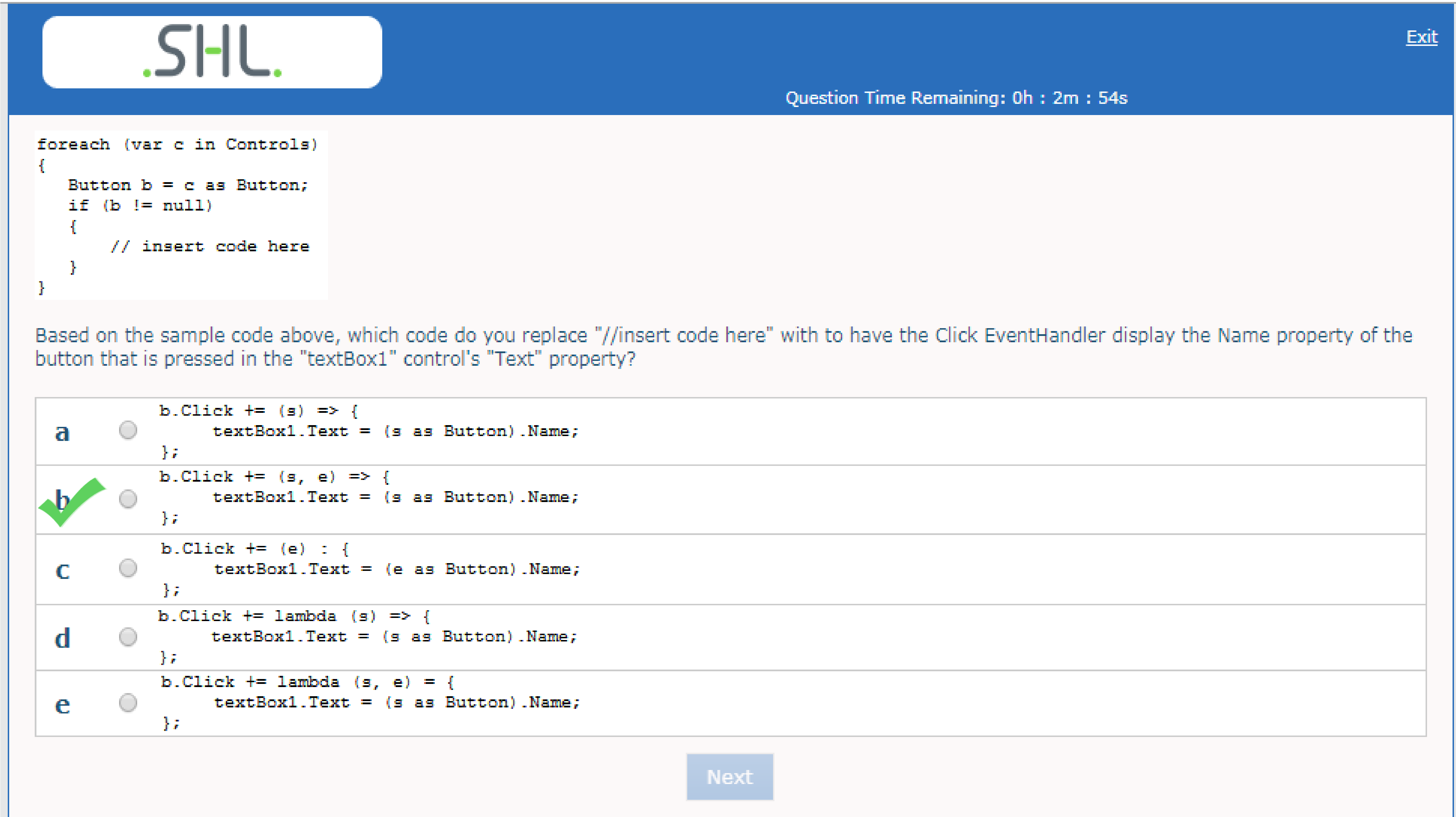
b
```
using MyDictionary = System.Collections.Generic.Dictionary<,>;
```

c
```
public class MyDictionary<TKey, TValue>  : Dictionary<TKey,
TValue>
   where TKey : Int32
{
}
```

d
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : ValueType
   where TValue : ReferenceType
{
}
```

✓ e
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : struct
   where TValue : class
{
}
```

Next



Exit

Question Time Remaining: 0h : 2m : 54s

```
foreach (var c in Controls)
{
   Button b = c as Button;
   if (b != null)
   {
       // insert code here
   }
}
```

Based on the sample code above, which code do you replace "//insert code here" with to have the Click EventHandler display the Name property of the button that is pressed in the "textBox1" control's "Text" property?

a
```
b.Click += (s) => {
     textBox1.Text = (s as Button).Name;
};
```

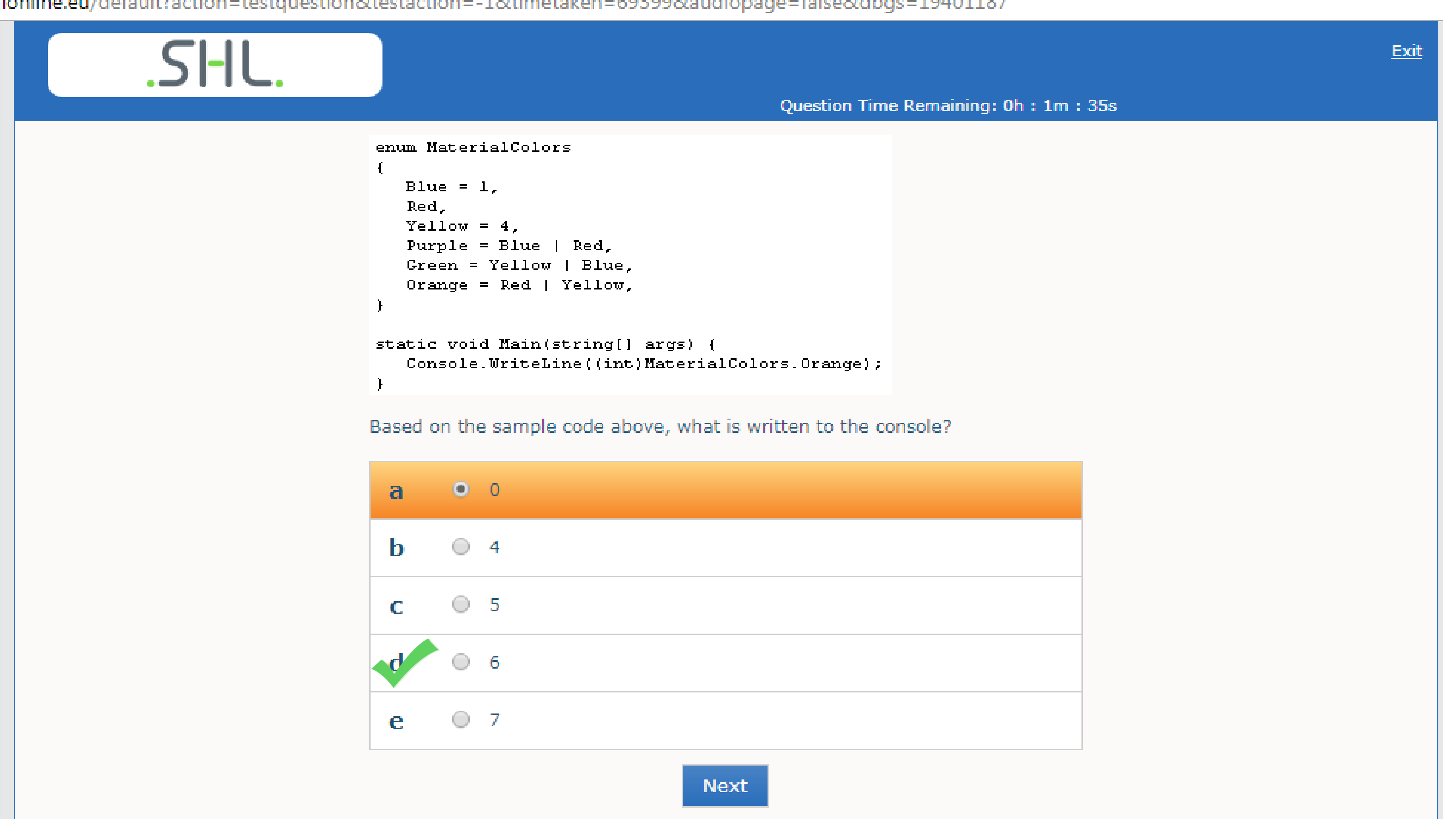
[✓ b]
```
b.Click += (s, e) => {
     textBox1.Text = (s as Button).Name;
};
```

c
```
b.Click += (e) : {
     textBox1.Text = (e as Button).Name;
};
```

d
```
b.Click += lambda (s) => {
     textBox1.Text = (s as Button).Name;
};
```

e
```
b.Click += lambda (s, e) = {
     textBox1.Text = (s as Button).Name;
};
```

Next



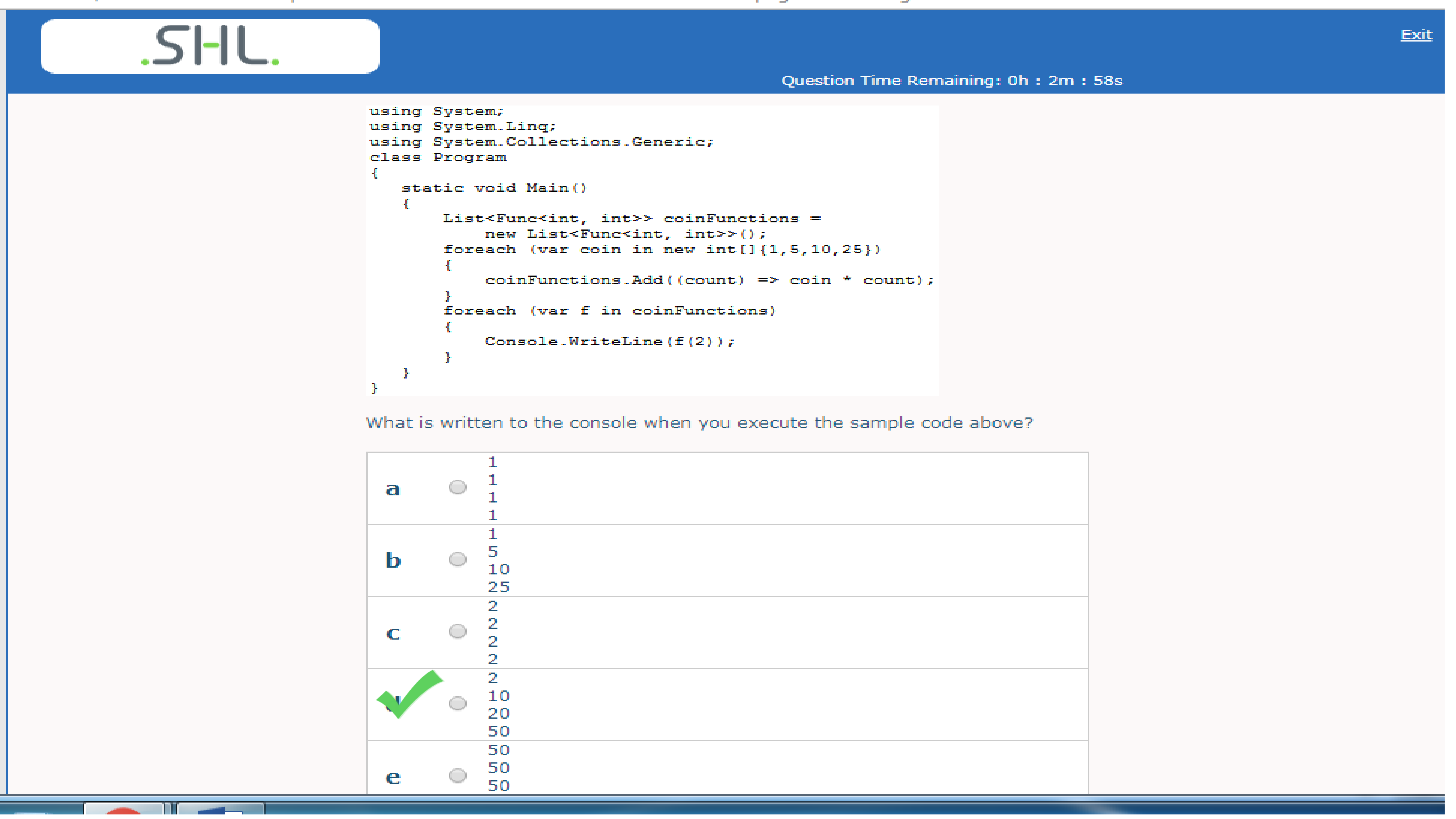
ionline.eu/default?action=testquestion&testaction=-1&timetaken=69399&audiopage=false&dbgs=19401187

Question Time Remaining: 0h : 1m : 35s

```
enum MaterialColors
{
   Blue = 1,
   Red,
   Yellow = 4,
   Purple = Blue | Red,
   Green = Yellow | Blue,
   Orange = Red | Yellow,
}

static void Main(string[] args) {
   Console.WriteLine((int)MaterialColors.Orange);
}
```

Based on the sample code above, what is written to the console?

a	●	0
b	○	4
c	○	5
✓ d	○	6
e	○	7



.SHL.

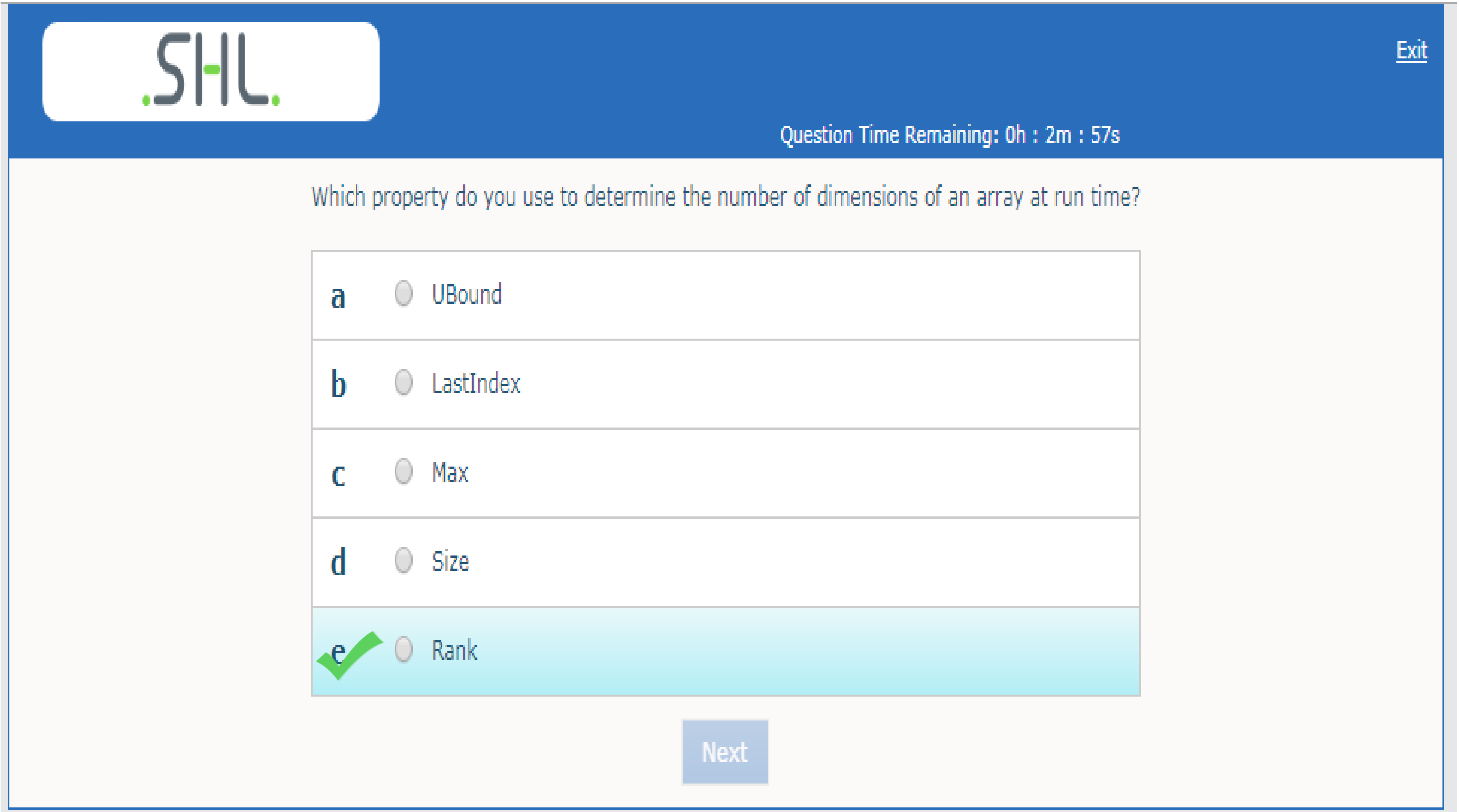
Exit

Question Time Remaining: 0h : 2m : 58s

```
using System;
using System.Linq;
using System.Collections.Generic;
class Program
{
    static void Main()
    {
        List<Func<int, int>> coinFunctions =
            new List<Func<int, int>>();
        foreach (var coin in new int[]{1,5,10,25})
        {
            coinFunctions.Add((count) => coin * count);
        }
        foreach (var f in coinFunctions)
        {
            Console.WriteLine(f(2));
        }
    }
}
```

What is written to the console when you execute the sample code above?

a	1 1 1 1
b	1 5 10 25
c	2 2 2 2
✓ d	2 10 20 50
e	50 50 50



Which property do you use to determine the number of dimensions of an array at run time?

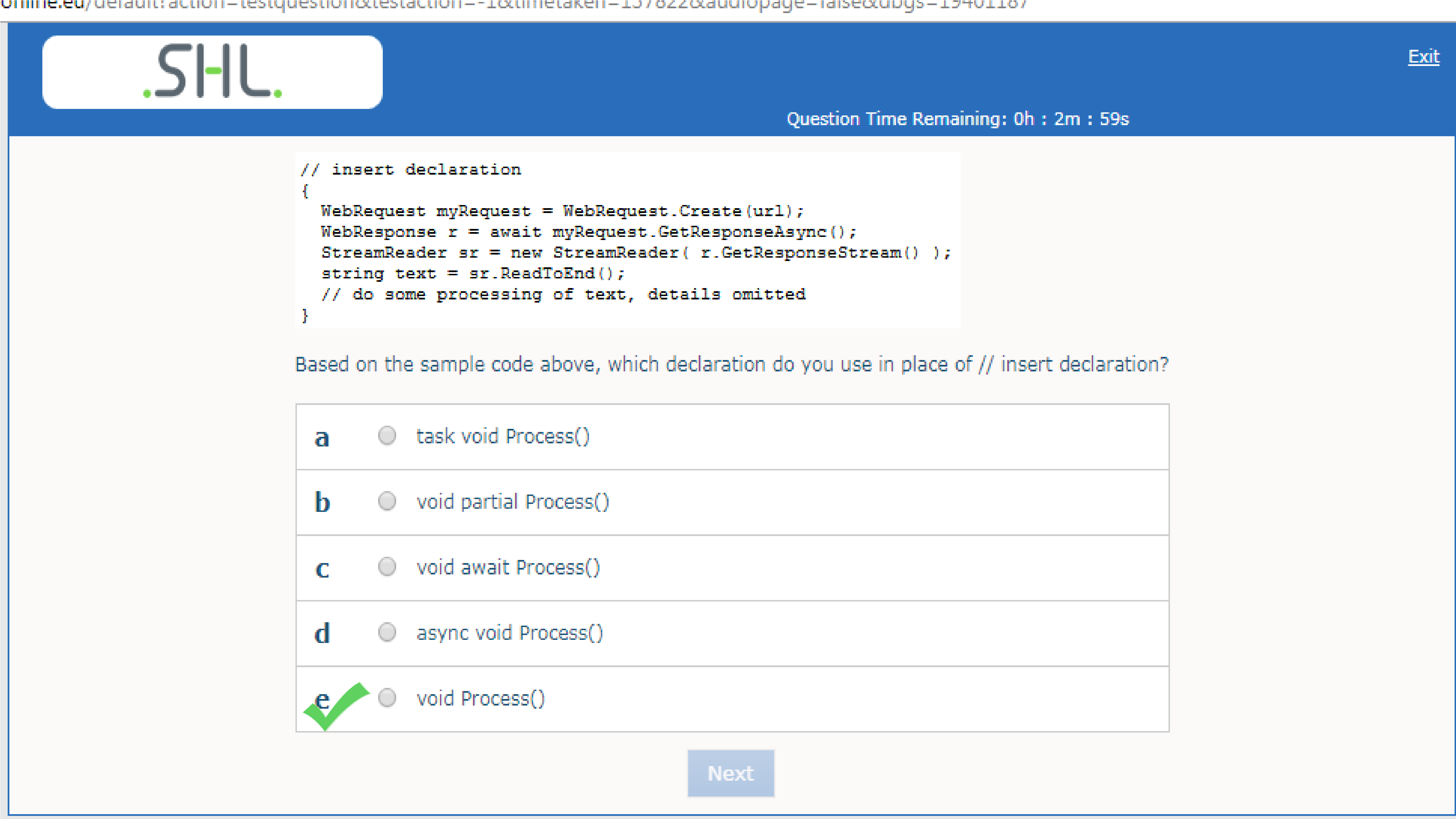
a UBound

b LastIndex

c Max

d Size

[✓ e] Rank



online.eu/default?action=testquestion&testaction=-1&timetaken=137822&audiopage=false&dbgs=19401187

Exit

Question Time Remaining: 0h : 2m : 59s

```
// insert declaration
{
  WebRequest myRequest = WebRequest.Create(url);
  WebResponse r = await myRequest.GetResponseAsync();
  StreamReader sr = new StreamReader( r.GetResponseStream() );
  string text = sr.ReadToEnd();
  // do some processing of text, details omitted
}
```

Based on the sample code above, which declaration do you use in place of // insert declaration?

a ○ task void Process()

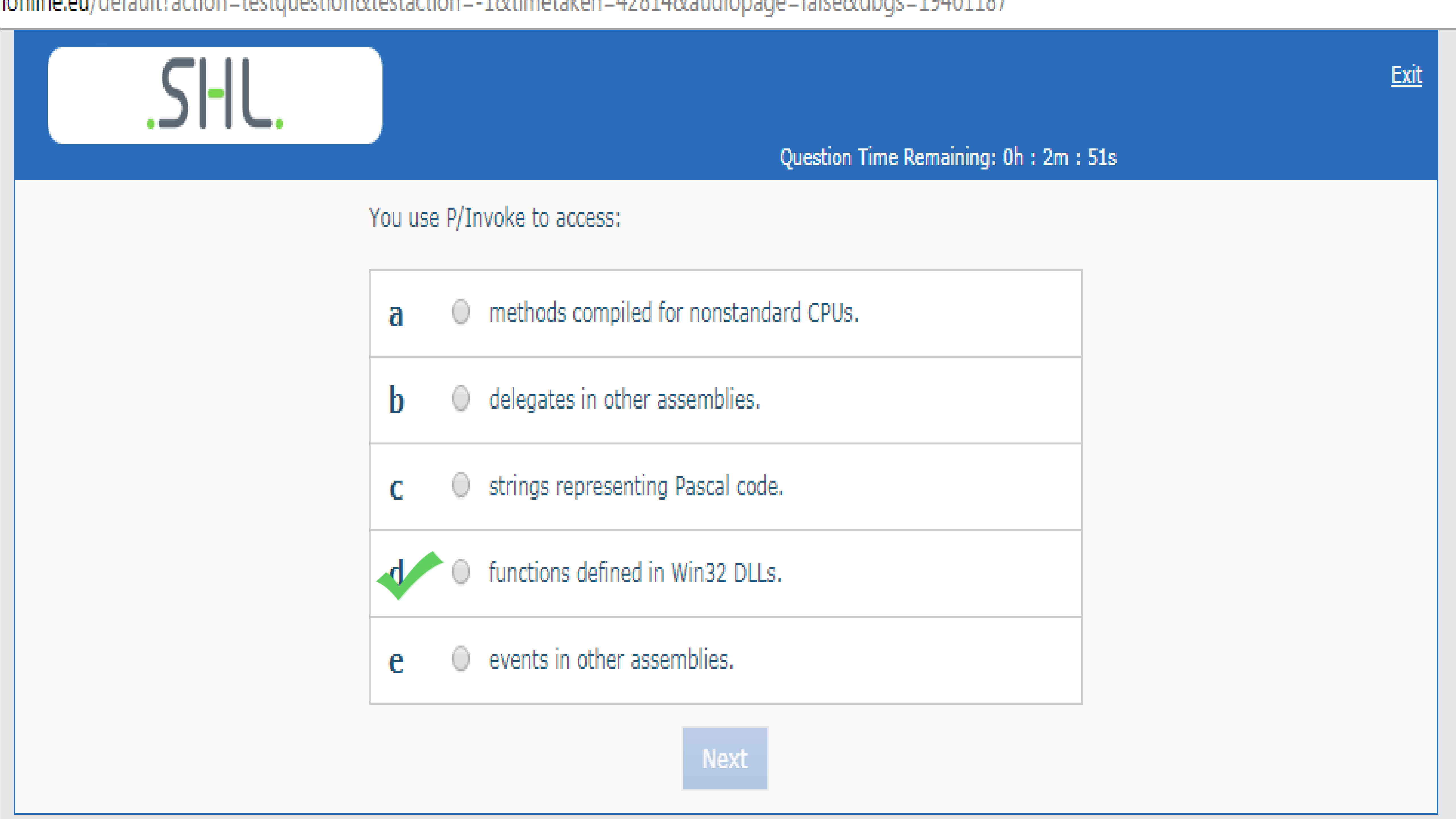
b ○ void partial Process()

c ○ void await Process()

d ○ async void Process()

[✓ e] ○ void Process()

Next


lonline.eu/default?action=testquestion&testaction=-1&timetaken=42814&audiopage=false&dbgs=19401187

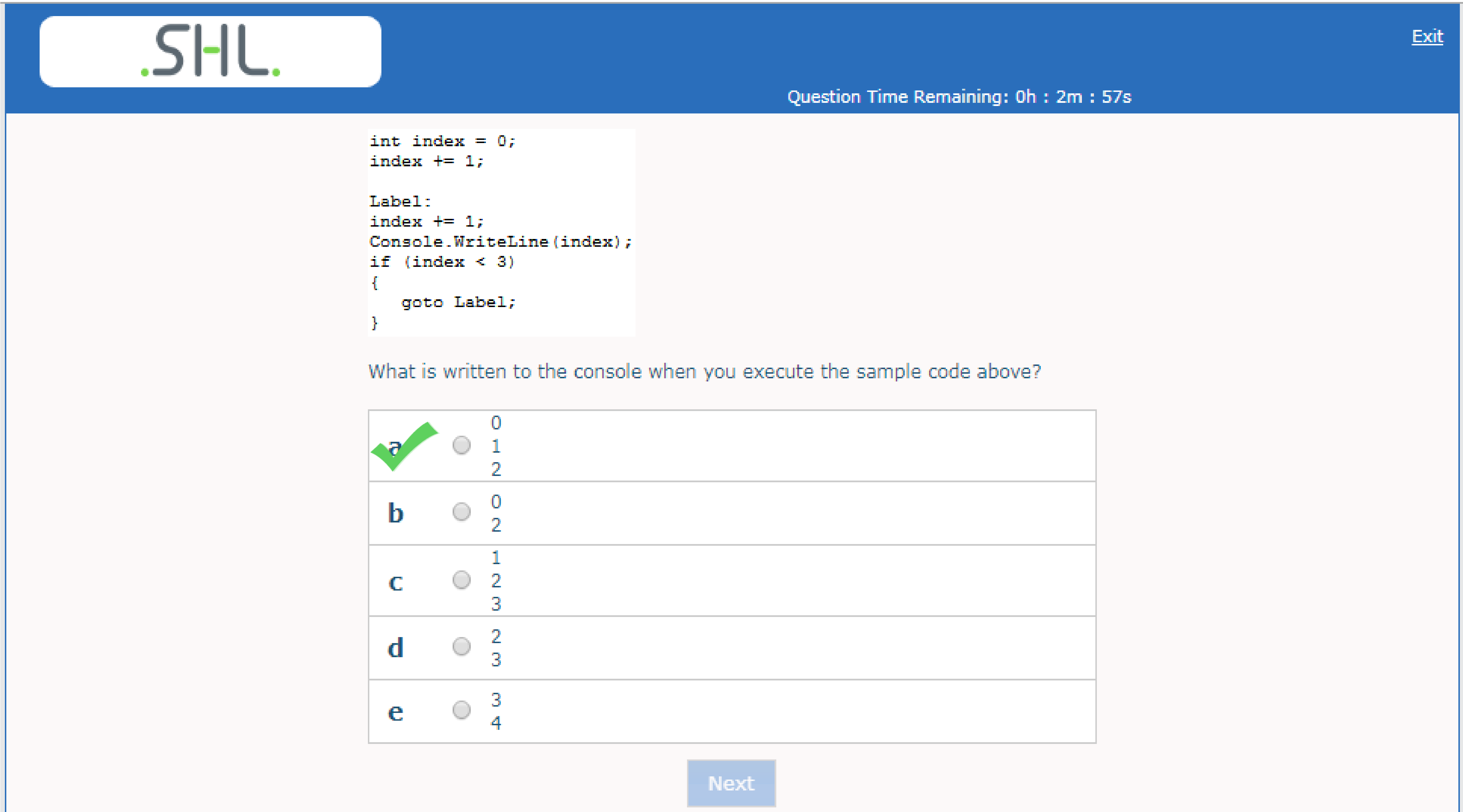
Exit

Question Time Remaining: 0h : 2m : 51s

You use P/Invoke to access:

a	methods compiled for nonstandard CPUs.
b	delegates in other assemblies.
c	strings representing Pascal code.
[✓ d]	functions defined in Win32 DLLs.
e	events in other assemblies.

Next



Exit

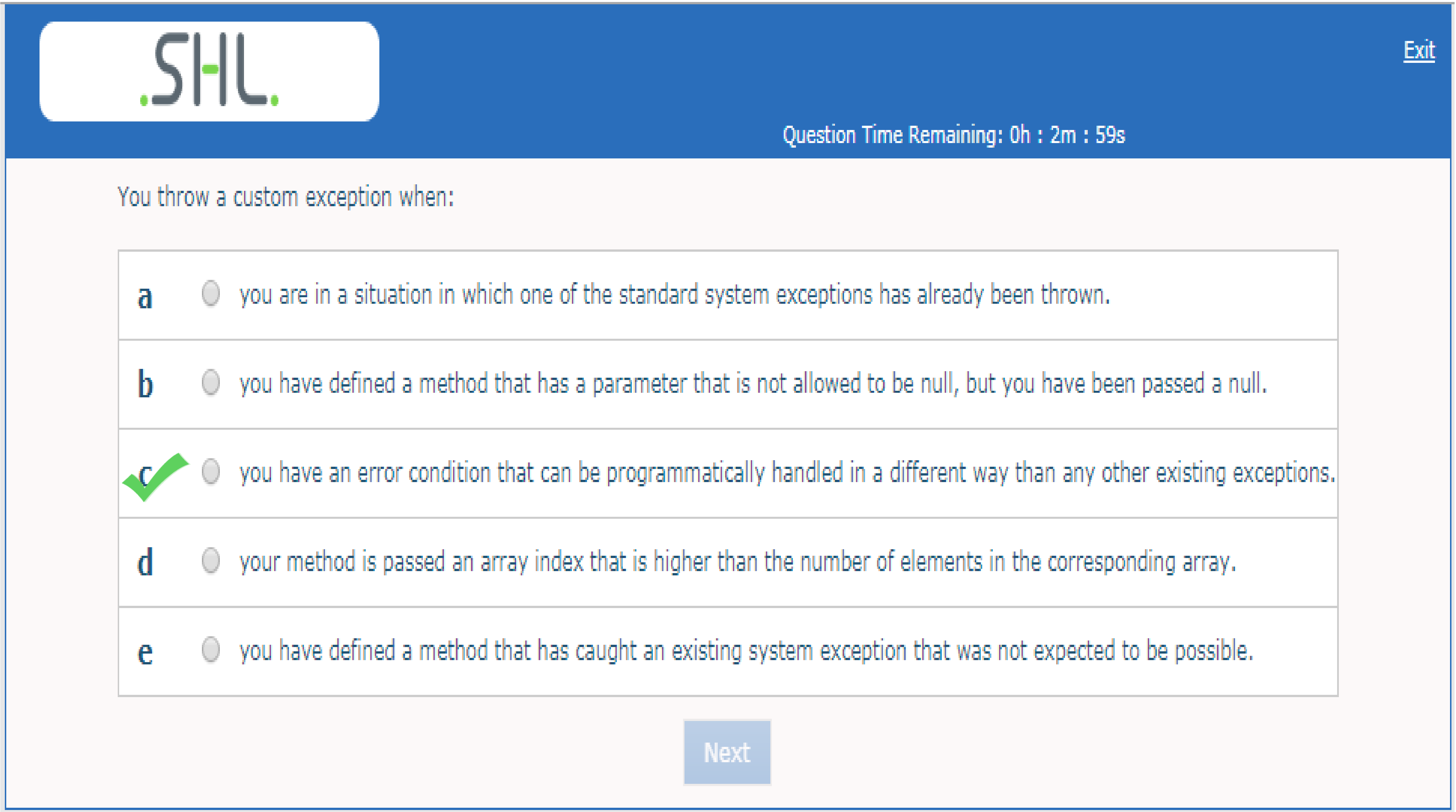
Question Time Remaining: 0h : 2m : 57s

```
int index = 0;
index += 1;

Label:
index += 1;
Console.WriteLine(index);
if (index < 3)
{
   goto Label;
}
```

What is written to the console when you execute the sample code above?

✓ a	○	0 1 2
b	○	0 2
c	○	1 2 3
d	○	2 3
e	○	3 4

Next

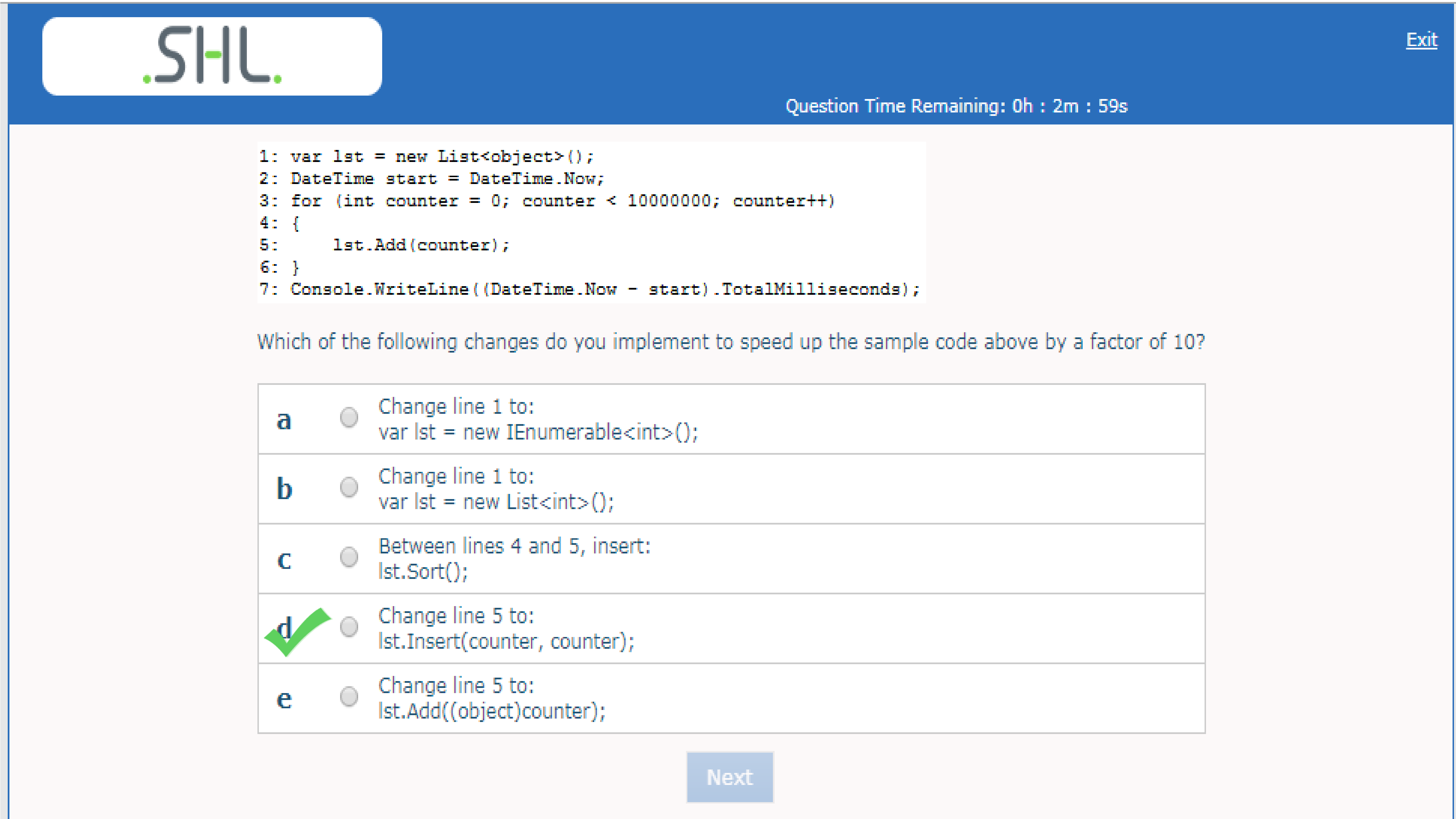

Exit

Question Time Remaining: 0h : 2m : 59s

You throw a custom exception when:

a	○	you are in a situation in which one of the standard system exceptions has already been thrown.
b	○	you have defined a method that has a parameter that is not allowed to be null, but you have been passed a null.
✓ c	○	you have an error condition that can be programmatically handled in a different way than any other existing exceptions.
d	○	your method is passed an array index that is higher than the number of elements in the corresponding array.
e	○	you have defined a method that has caught an existing system exception that was not expected to be possible.

Next

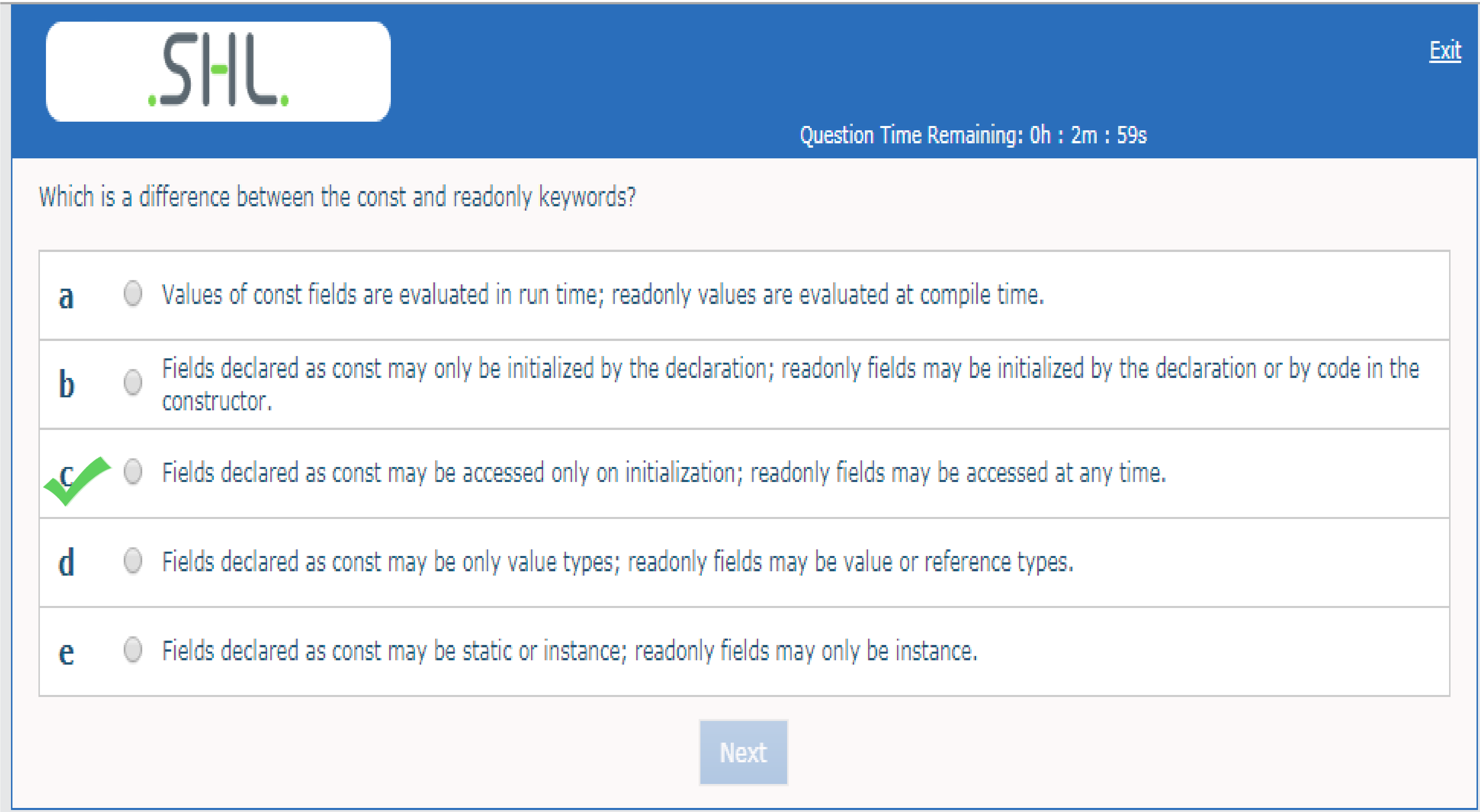


Question Time Remaining: 0h : 2m : 59s

```
1: var lst = new List<object>();
2: DateTime start = DateTime.Now;
3: for (int counter = 0; counter < 10000000; counter++)
4: {
5:     lst.Add(counter);
6: }
7: Console.WriteLine((DateTime.Now - start).TotalMilliseconds);
```

Which of the following changes do you implement to speed up the sample code above by a factor of 10?

a	Change line 1 to: var lst = new IEnumerable<int>();
b	Change line 1 to: var lst = new List<int>();
c	Between lines 4 and 5, insert: lst.Sort();
[✓ d]	Change line 5 to: lst.Insert(counter, counter);
e	Change line 5 to: lst.Add((object)counter);

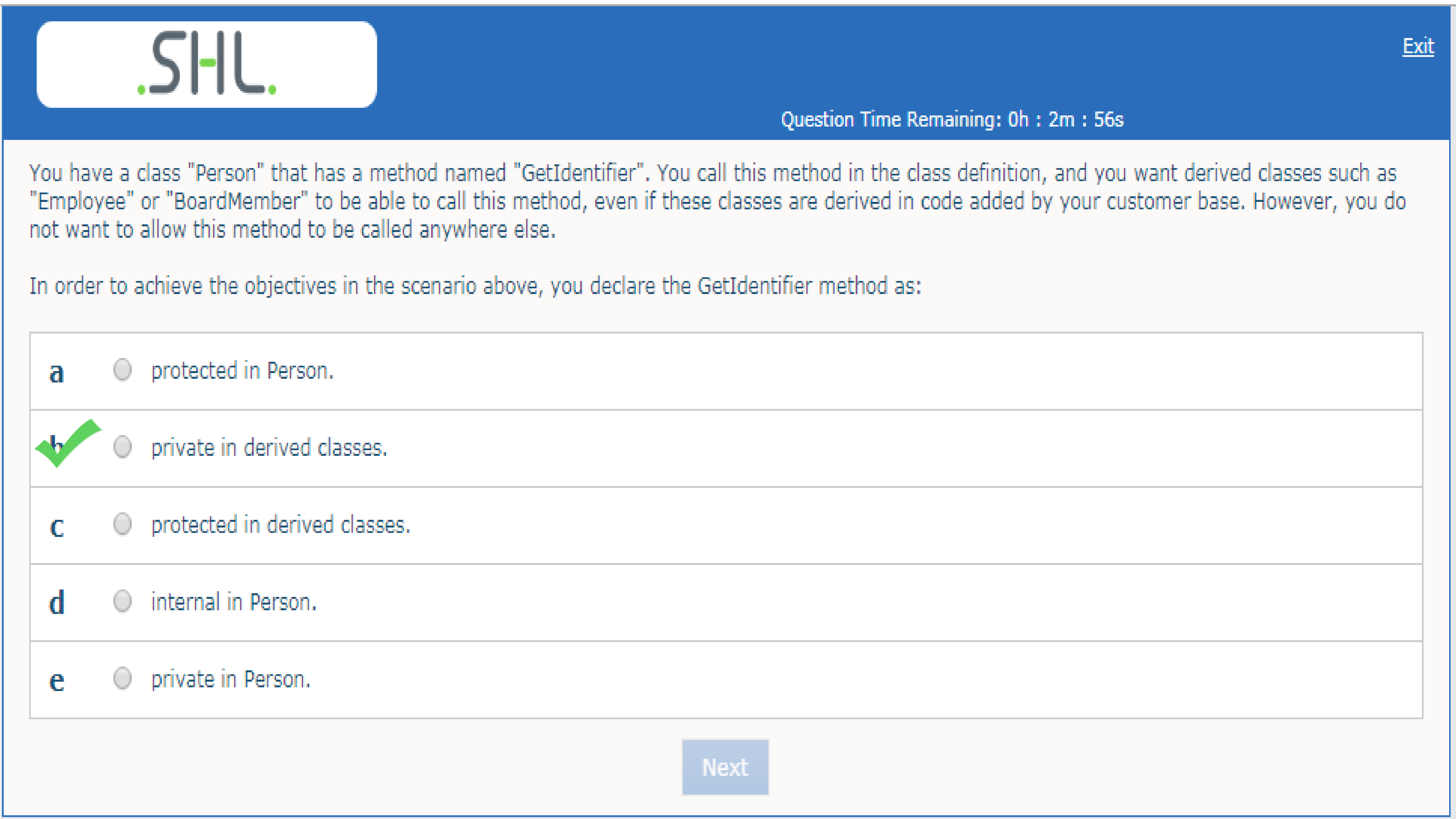

Exit

Question Time Remaining: 0h : 2m : 59s

Which is a difference between the const and readonly keywords?

a	Values of const fields are evaluated in run time; readonly values are evaluated at compile time.
b	Fields declared as const may only be initialized by the declaration; readonly fields may be initialized by the declaration or by code in the constructor.
✓ c	Fields declared as const may be accessed only on initialization; readonly fields may be accessed at any time.
d	Fields declared as const may be only value types; readonly fields may be value or reference types.
e	Fields declared as const may be static or instance; readonly fields may only be instance.

Next

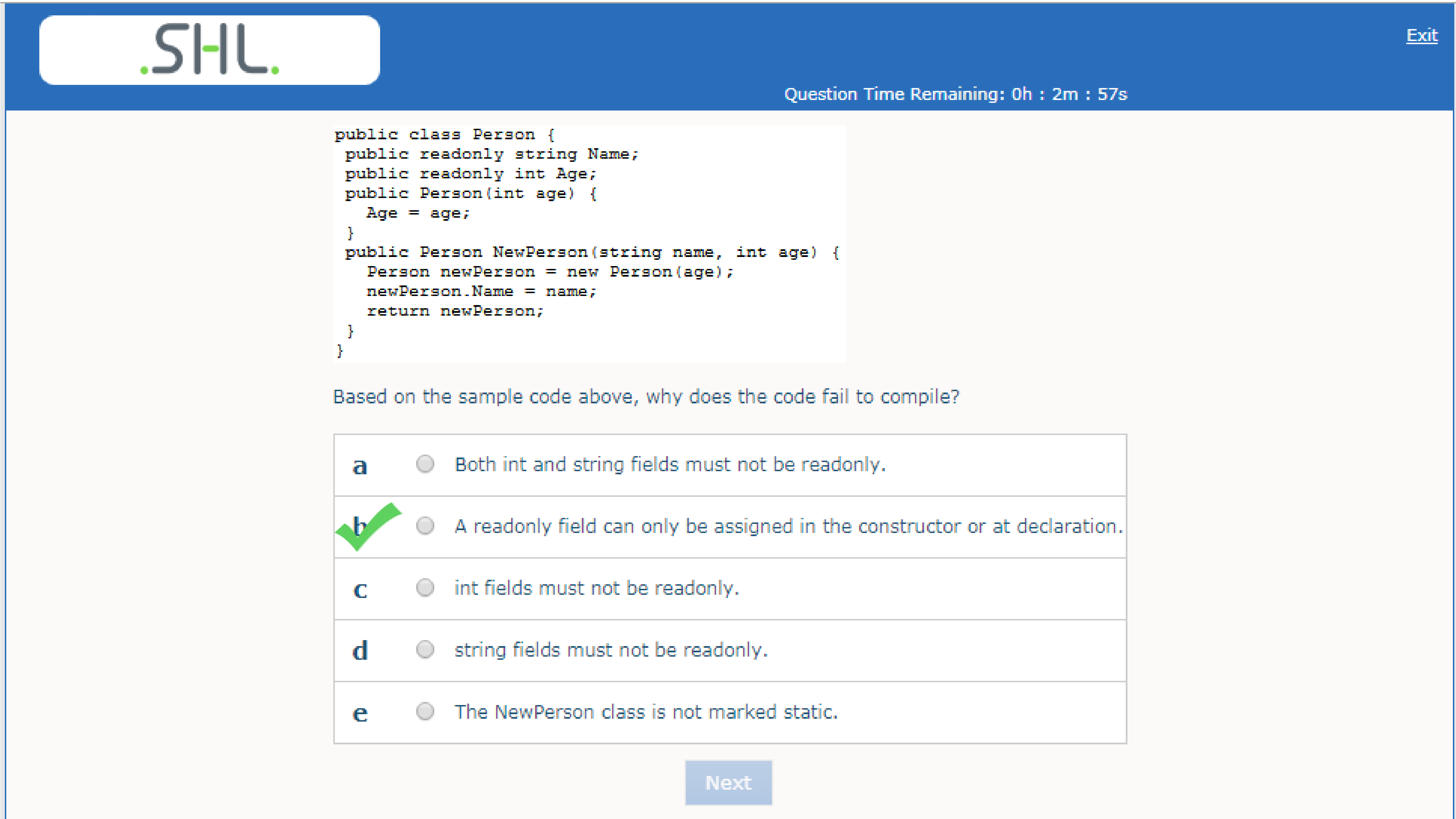


Question Time Remaining: 0h : 2m : 56s

You have a class "Person" that has a method named "GetIdentifier". You call this method in the class definition, and you want derived classes such as "Employee" or "BoardMember" to be able to call this method, even if these classes are derived in code added by your customer base. However, you do not want to allow this method to be called anywhere else.

In order to achieve the objectives in the scenario above, you declare the GetIdentifier method as:

a	protected in Person.
[✓ b]	private in derived classes.
c	protected in derived classes.
d	internal in Person.
e	private in Person.



Exit

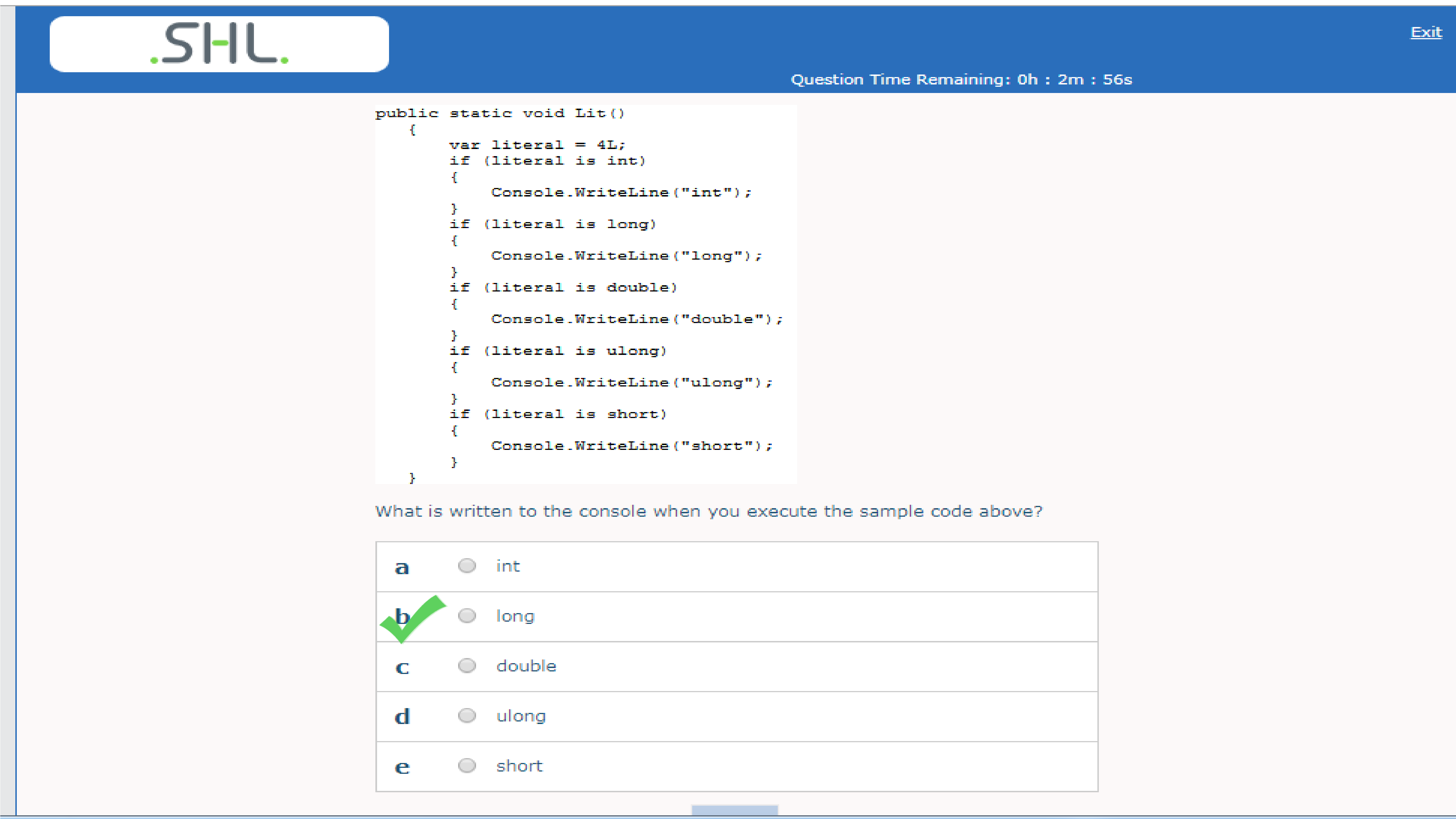
Question Time Remaining: 0h : 2m : 57s

```
public class Person {
 public readonly string Name;
 public readonly int Age;
 public Person(int age) {
   Age = age;
 }
 public Person NewPerson(string name, int age) {
   Person newPerson = new Person(age);
   newPerson.Name = name;
   return newPerson;
 }
}
```

Based on the sample code above, why does the code fail to compile?

a	Both int and string fields must not be readonly.
[✓ b]	A readonly field can only be assigned in the constructor or at declaration.
c	int fields must not be readonly.
d	string fields must not be readonly.
e	The NewPerson class is not marked static.

Next

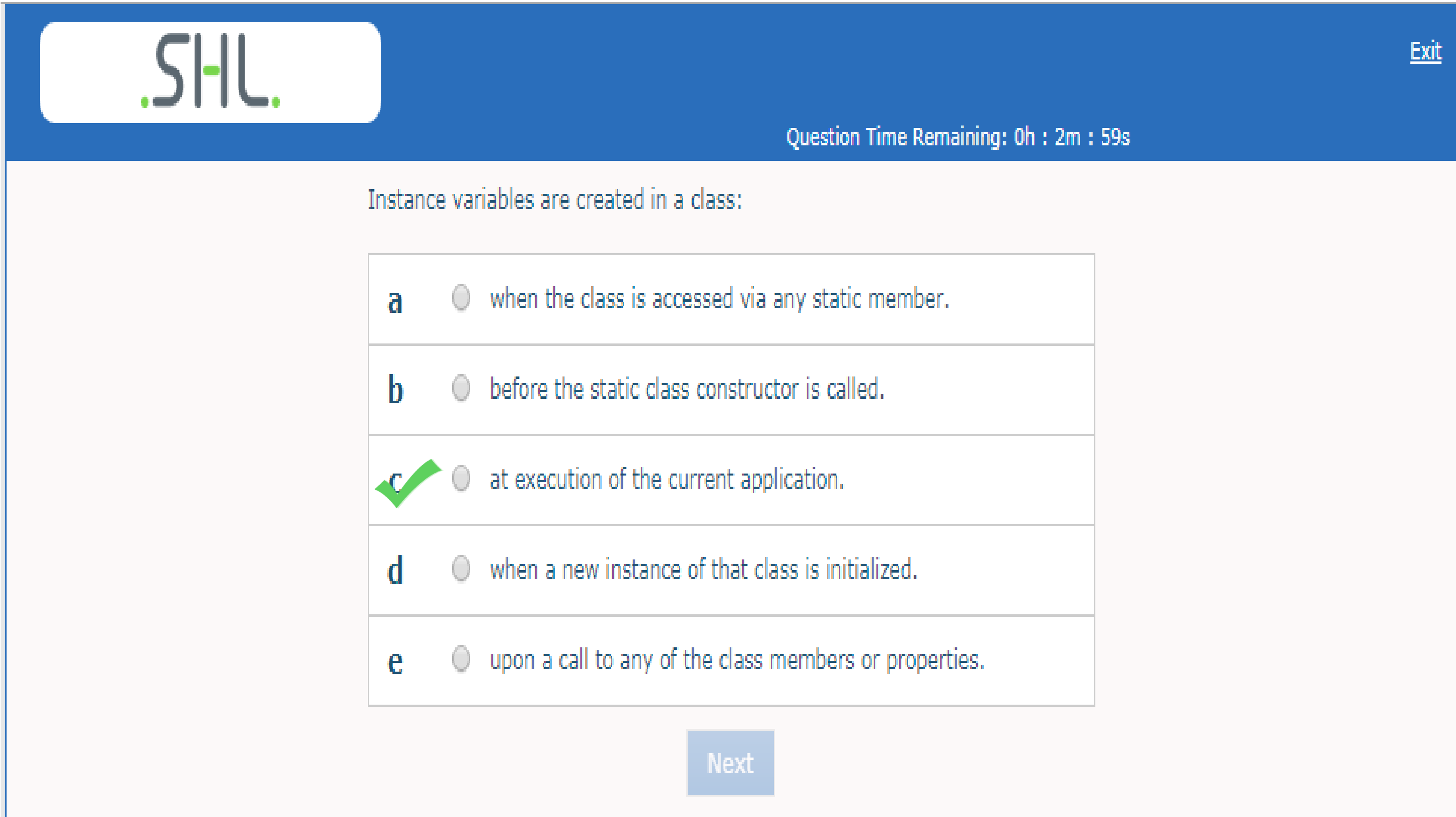


Question Time Remaining: 0h : 2m : 56s

```
public static void Lit()
    {
        var literal = 4L;
        if (literal is int)
        {
            Console.WriteLine("int");
        }
        if (literal is long)
        {
            Console.WriteLine("long");
        }
        if (literal is double)
        {
            Console.WriteLine("double");
        }
        if (literal is ulong)
        {
            Console.WriteLine("ulong");
        }
        if (literal is short)
        {
            Console.WriteLine("short");
        }
    }
```

What is written to the console when you execute the sample code above?

a	int
[✓ b]	long
c	double
d	ulong
e	short



Exit

Question Time Remaining: 0h : 2m : 59s

Instance variables are created in a class:

a ○ when the class is accessed via any static member.

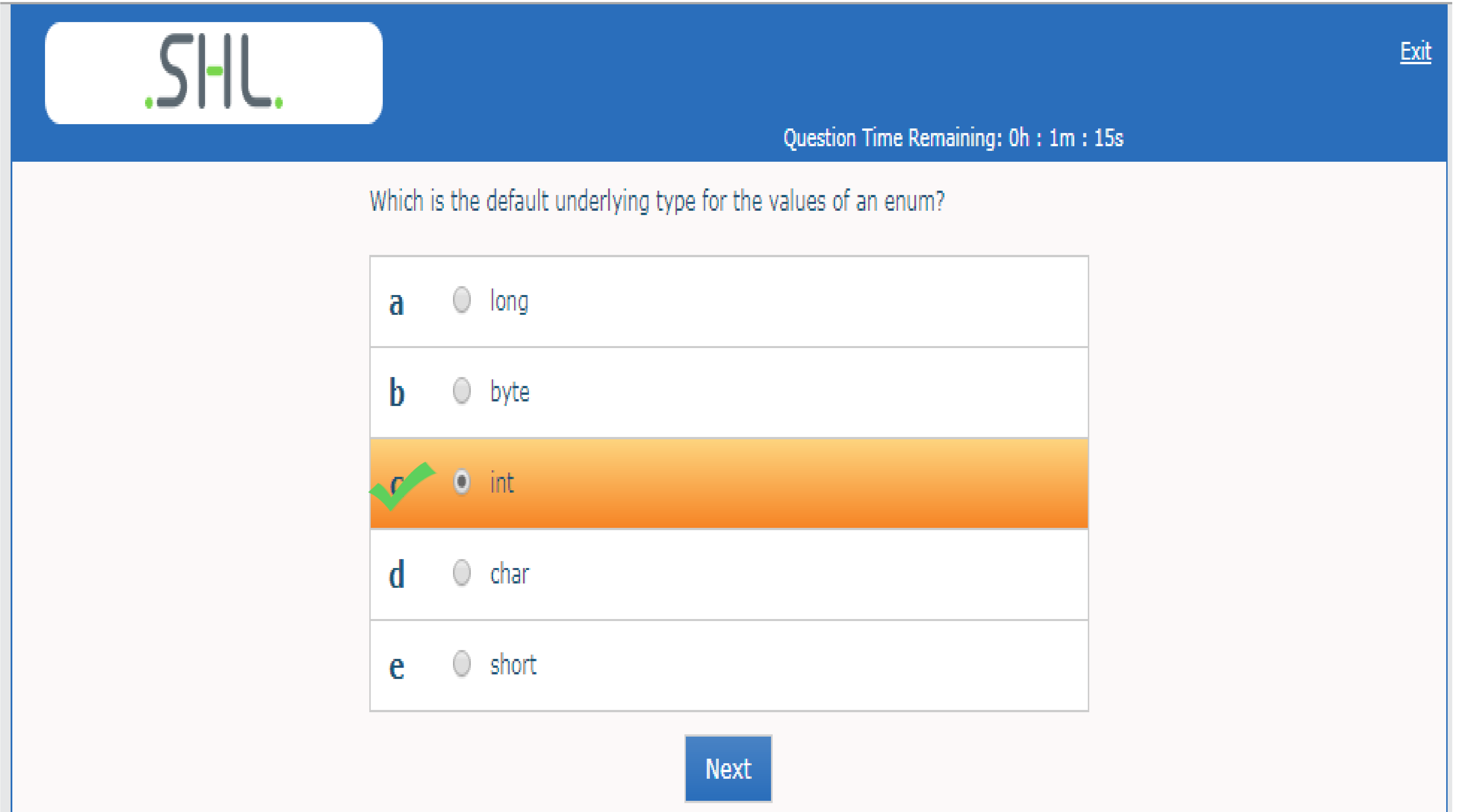
b ○ before the static class constructor is called.

✓ c ○ at execution of the current application.

d ○ when a new instance of that class is initialized.

e ○ upon a call to any of the class members or properties.

Next



.SHL.

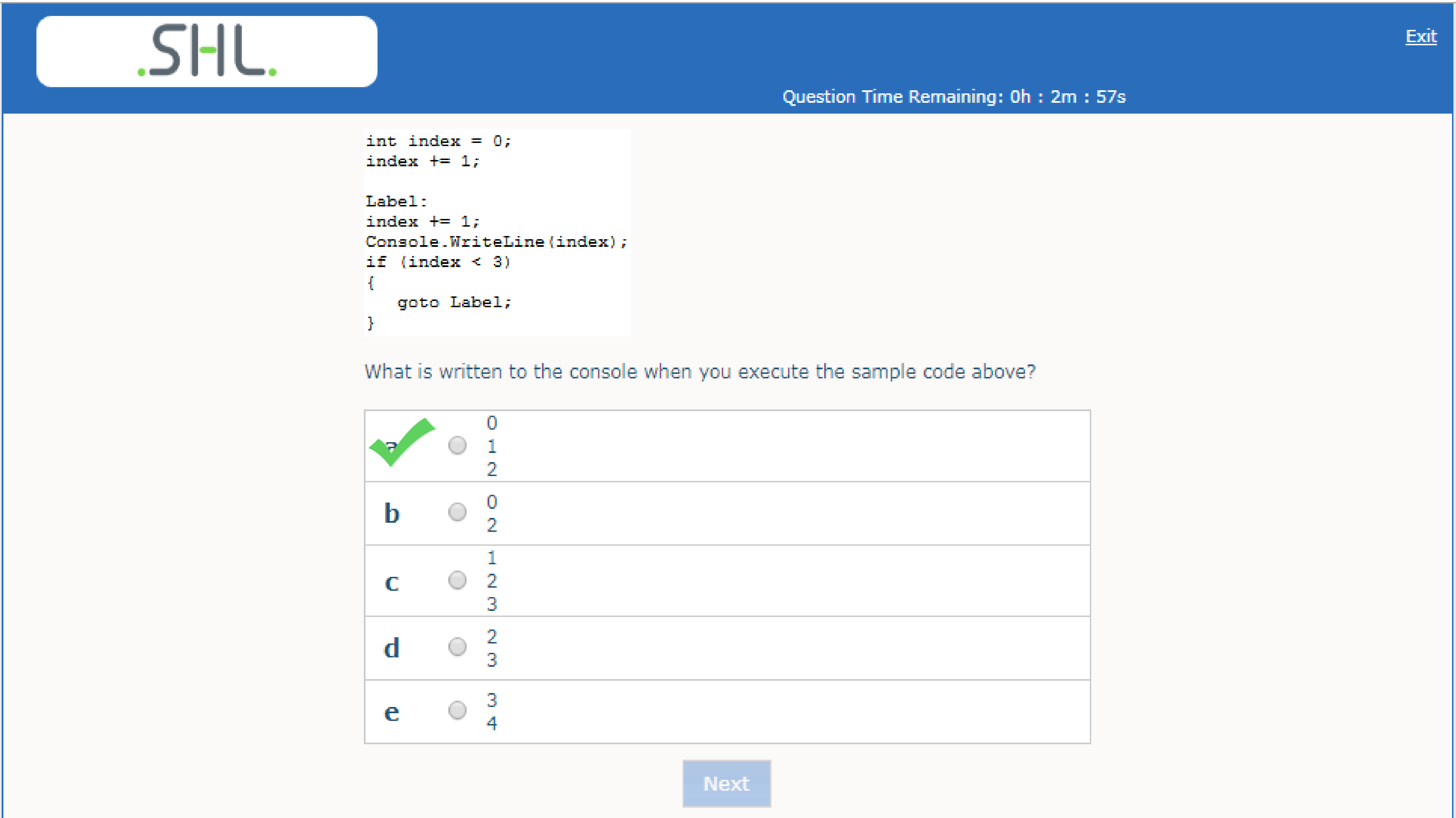
Exit

Question Time Remaining: 0h : 1m : 15s

Which is the default underlying type for the values of an enum?

a long
b byte
[✓ c int]
d char
e short

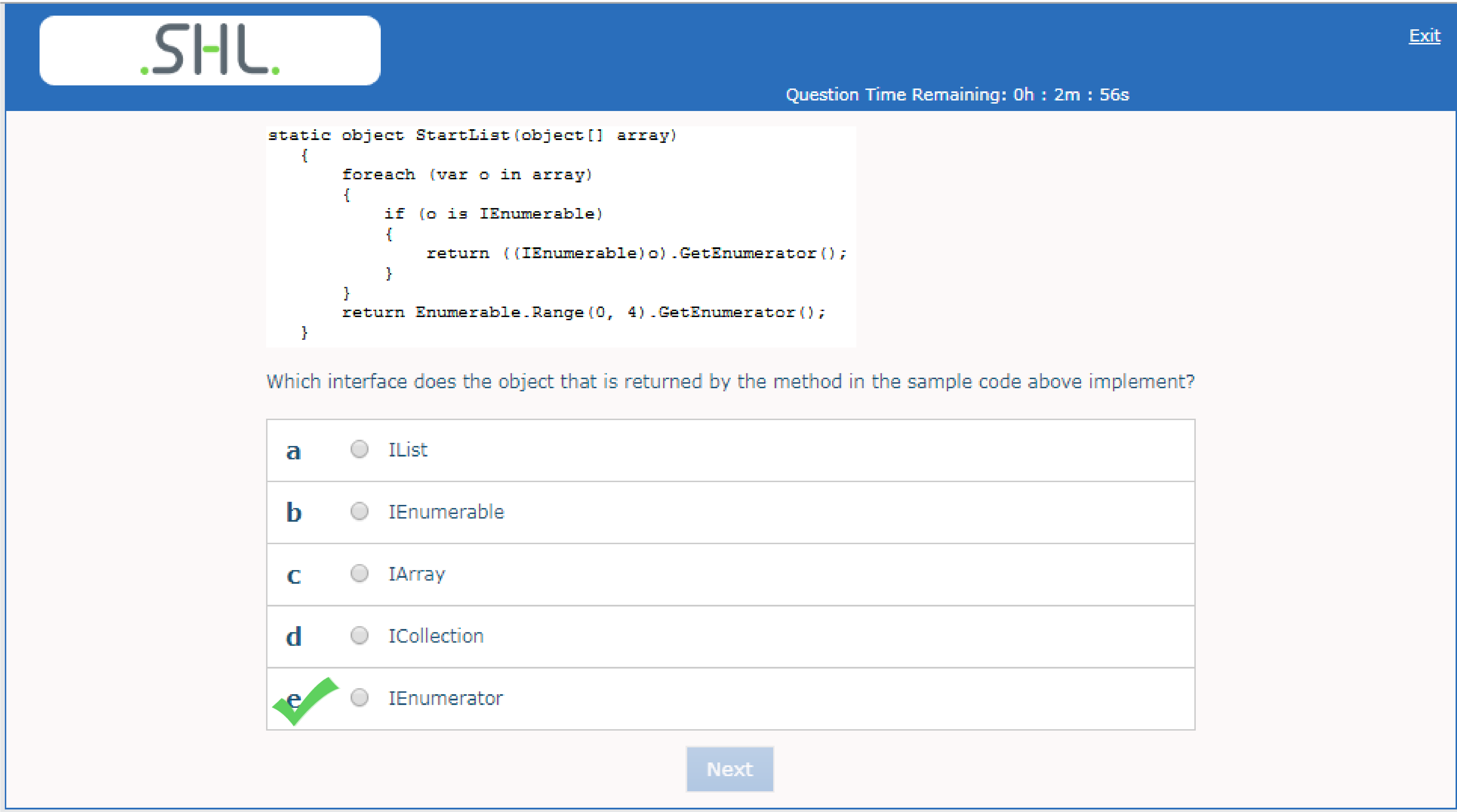
Next



```
int index = 0;
index += 1;

Label:
index += 1;
Console.WriteLine(index);
if (index < 3)
{
   goto Label;
}
```

What is written to the console when you execute the sample code above?

[✓ a]	○	0 1 2
b	○	0 2
c	○	1 2 3
d	○	2 3
e	○	3 4



Exit

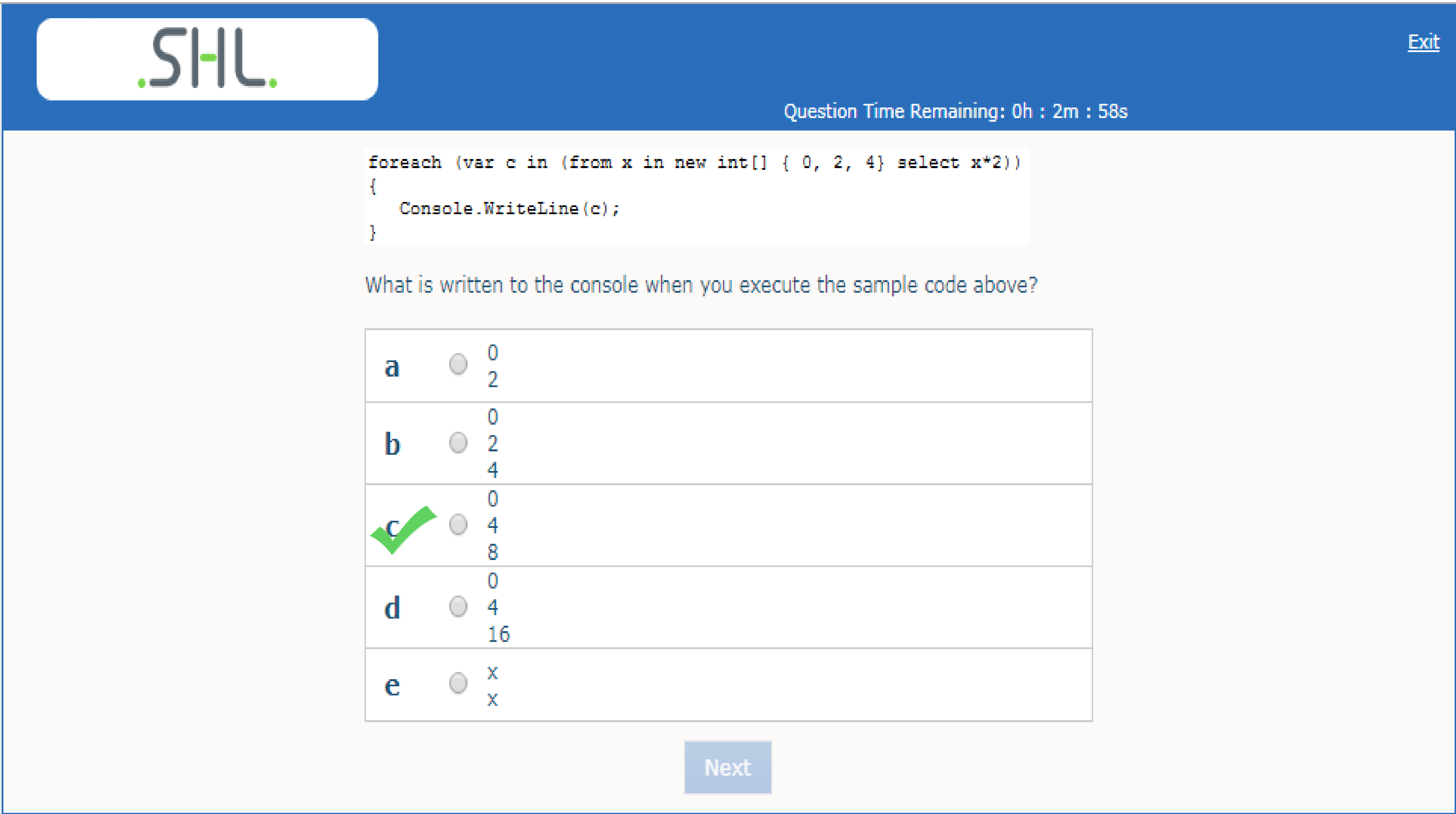
Question Time Remaining: 0h : 2m : 56s

```
static object StartList(object[] array)
   {
       foreach (var o in array)
       {
           if (o is IEnumerable)
           {
               return ((IEnumerable)o).GetEnumerator();
           }
       }
       return Enumerable.Range(0, 4).GetEnumerator();
   }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	IList
b	IEnumerable
c	IArray
d	ICollection
[✓ e]	IEnumerator

Next

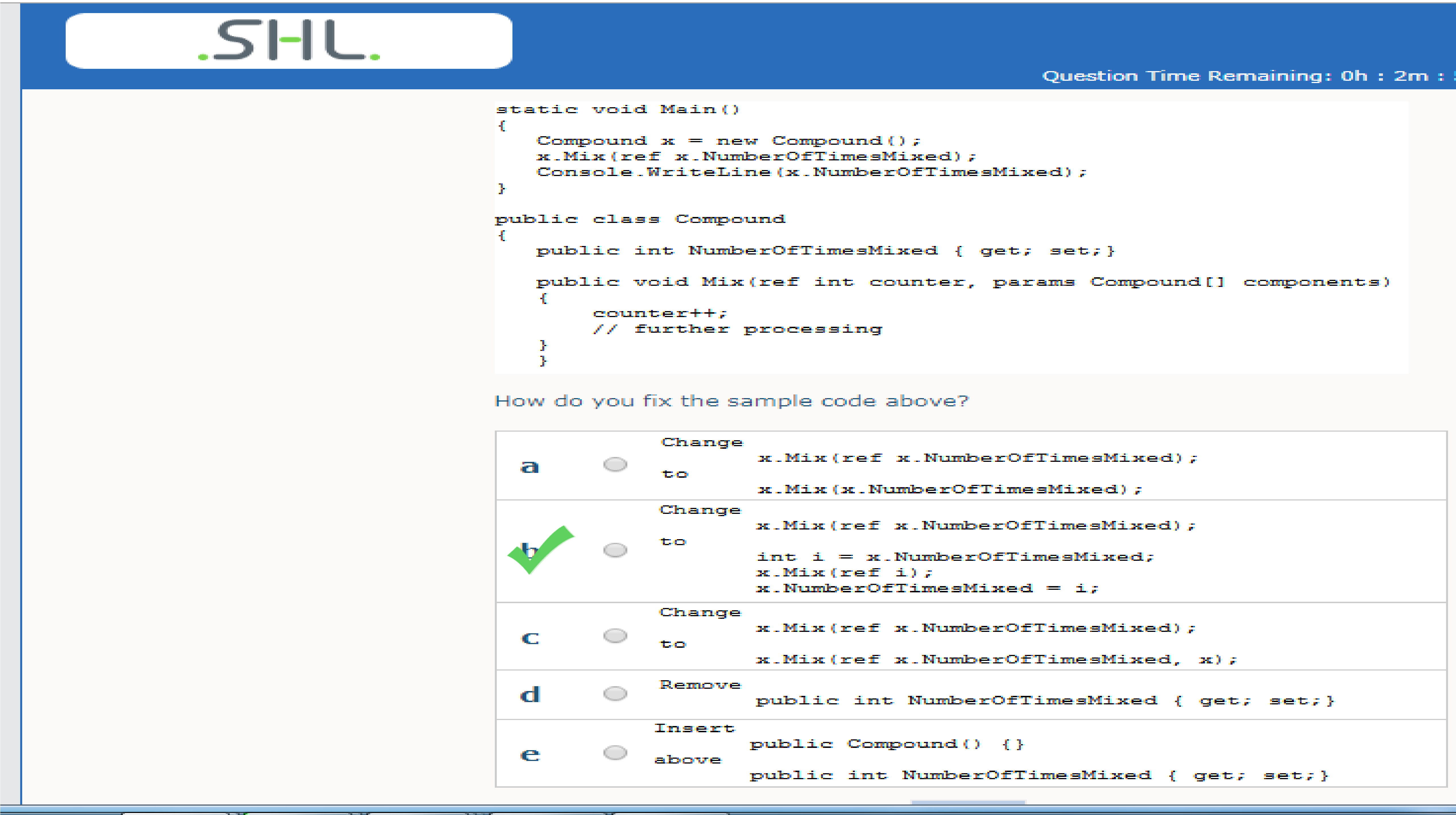


Question Time Remaining: 0h : 2m : 58s

```
foreach (var c in (from x in new int[] { 0, 2, 4} select x*2))
{
   Console.WriteLine(c);
}
```

What is written to the console when you execute the sample code above?

a	0 2
b	0 2 4
[✓ c]	0 4 8
d	0 4 16
e	x x

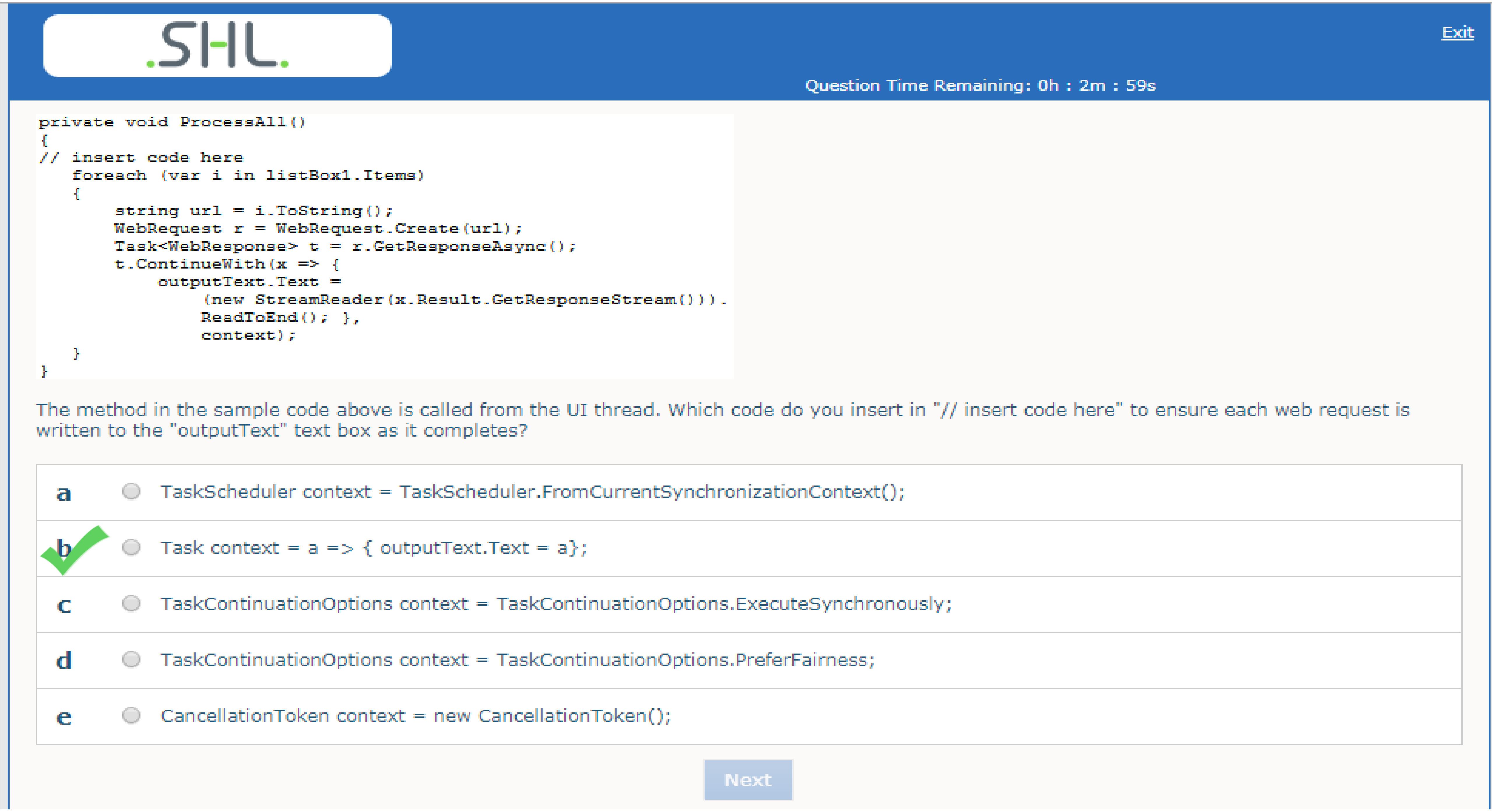

.SHL.

Question Time Remaining: 0h : 2m :

```
static void Main()
{
    Compound x = new Compound();
    x.Mix(ref x.NumberOfTimesMixed);
    Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
    public int NumberOfTimesMixed { get; set;}

    public void Mix(ref int counter, params Compound[] components)
    {
        counter++;
        // further processing
    }
    }
```

How do you fix the sample code above?

a	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(x.NumberOfTimesMixed);`
[✓ b]	Change `x.Mix(ref x.NumberOfTimesMixed);` to `int i = x.NumberOfTimesMixed;` `x.Mix(ref i);` `x.NumberOfTimesMixed = i;`
c	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(ref x.NumberOfTimesMixed, x);`
d	Remove `public int NumberOfTimesMixed { get; set;}`
e	Insert `public Compound() {}` above `public int NumberOfTimesMixed { get; set;}`



SHL

Exit

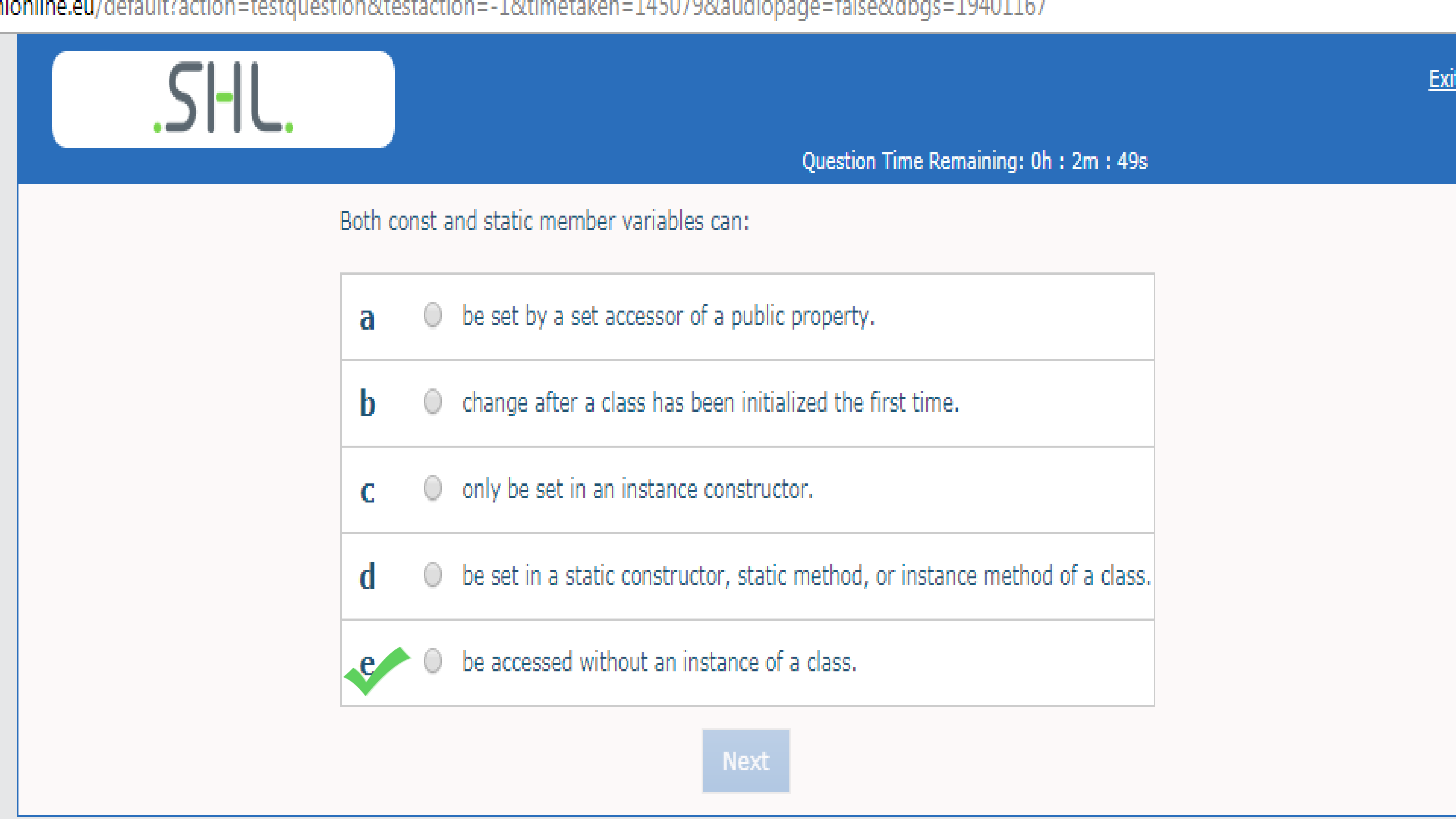
Question Time Remaining: 0h : 2m : 59s

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

a	TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();
[✓ b]	Task context = a => { outputText.Text = a};
c	TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;
d	TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;
e	CancellationToken context = new CancellationToken();

Next



lonline.eu/default?action=testquestion&testaction=-1&timetaken=145079&audiopage=false&dbgs=19401167

.SHL.

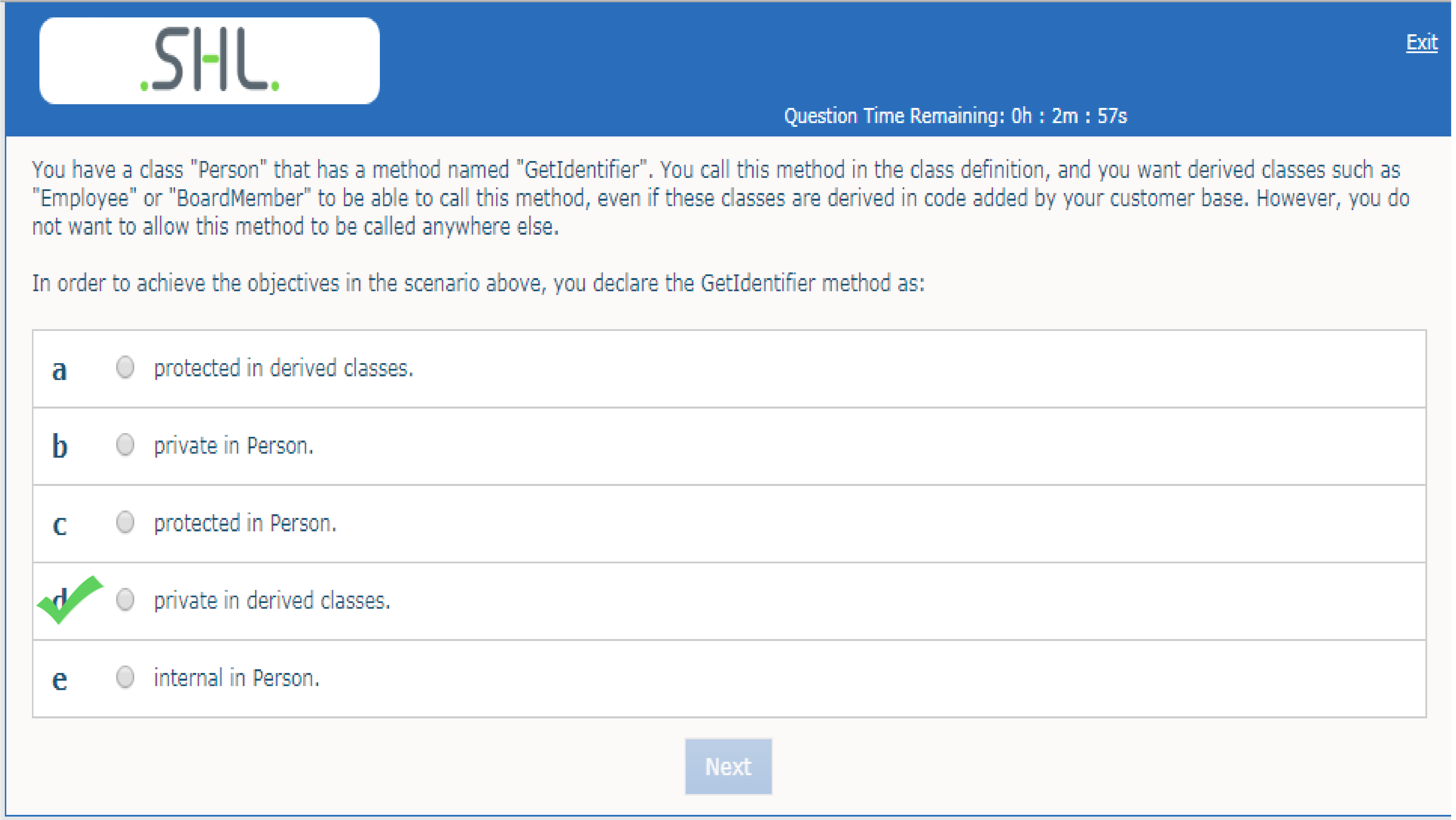
Exi

Question Time Remaining: 0h : 2m : 49s

Both const and static member variables can:

a	be set by a set accessor of a public property.
b	change after a class has been initialized the first time.
c	only be set in an instance constructor.
d	be set in a static constructor, static method, or instance method of a class.
[✓ e]	be accessed without an instance of a class.

Next



SHL.

Exit

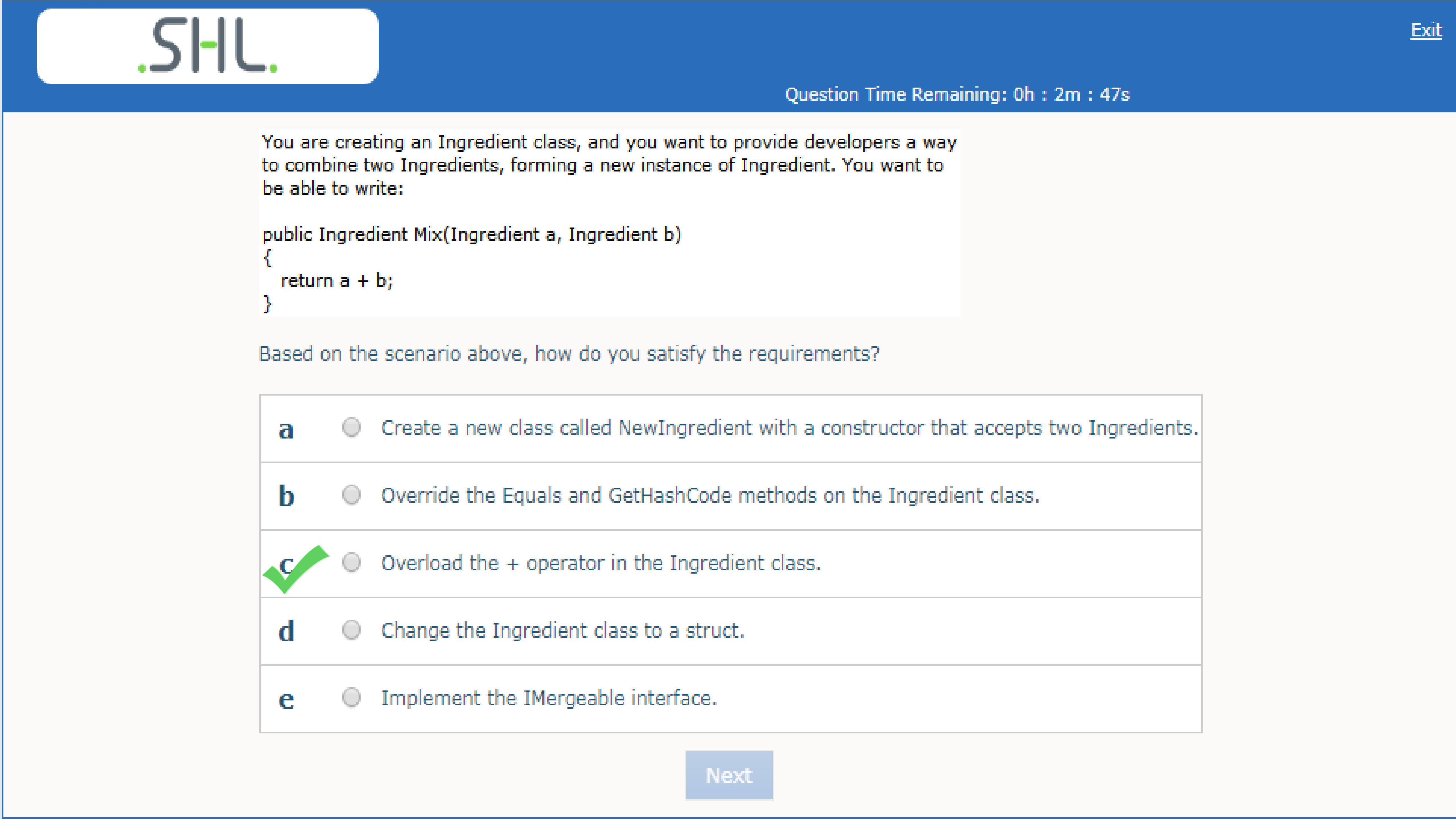
Question Time Remaining: 0h : 2m : 57s

You have a class "Person" that has a method named "GetIdentifier". You call this method in the class definition, and you want derived classes such as "Employee" or "BoardMember" to be able to call this method, even if these classes are derived in code added by your customer base. However, you do not want to allow this method to be called anywhere else.

In order to achieve the objectives in the scenario above, you declare the GetIdentifier method as:

a	○	protected in derived classes.
b	○	private in Person.
c	○	protected in Person.
✓ d	○	private in derived classes.
e	○	internal in Person.

Next



Question Time Remaining: 0h : 2m : 47s

You are creating an Ingredient class, and you want to provide developers a way to combine two Ingredients, forming a new instance of Ingredient. You want to be able to write:

```
public Ingredient Mix(Ingredient a, Ingredient b)
{
  return a + b;
}
```

Based on the scenario above, how do you satisfy the requirements?

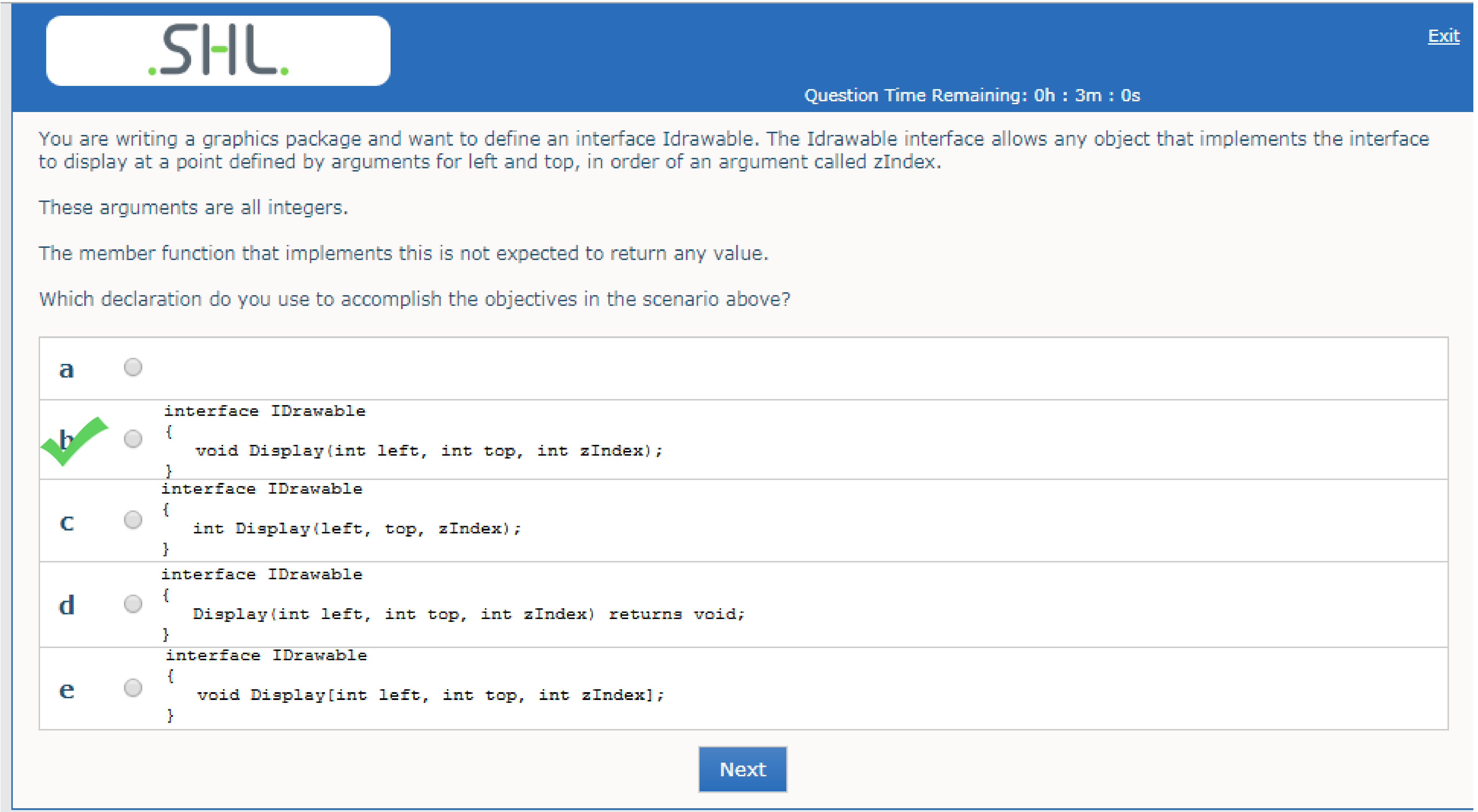
a Create a new class called NewIngredient with a constructor that accepts two Ingredients.

b Override the Equals and GetHashCode methods on the Ingredient class.

[✓ c] Overload the + operator in the Ingredient class.

d Change the Ingredient class to a struct.

e Implement the IMergeable interface.



You are writing a graphics package and want to define an interface Idrawable. The Idrawable interface allows any object that implements the interface to display at a point defined by arguments for left and top, in order of an argument called zIndex.

These arguments are all integers.

The member function that implements this is not expected to return any value.

Which declaration do you use to accomplish the objectives in the scenario above?

a

✓ b

```
interface IDrawable
{
   void Display(int left, int top, int zIndex);
}
```

c

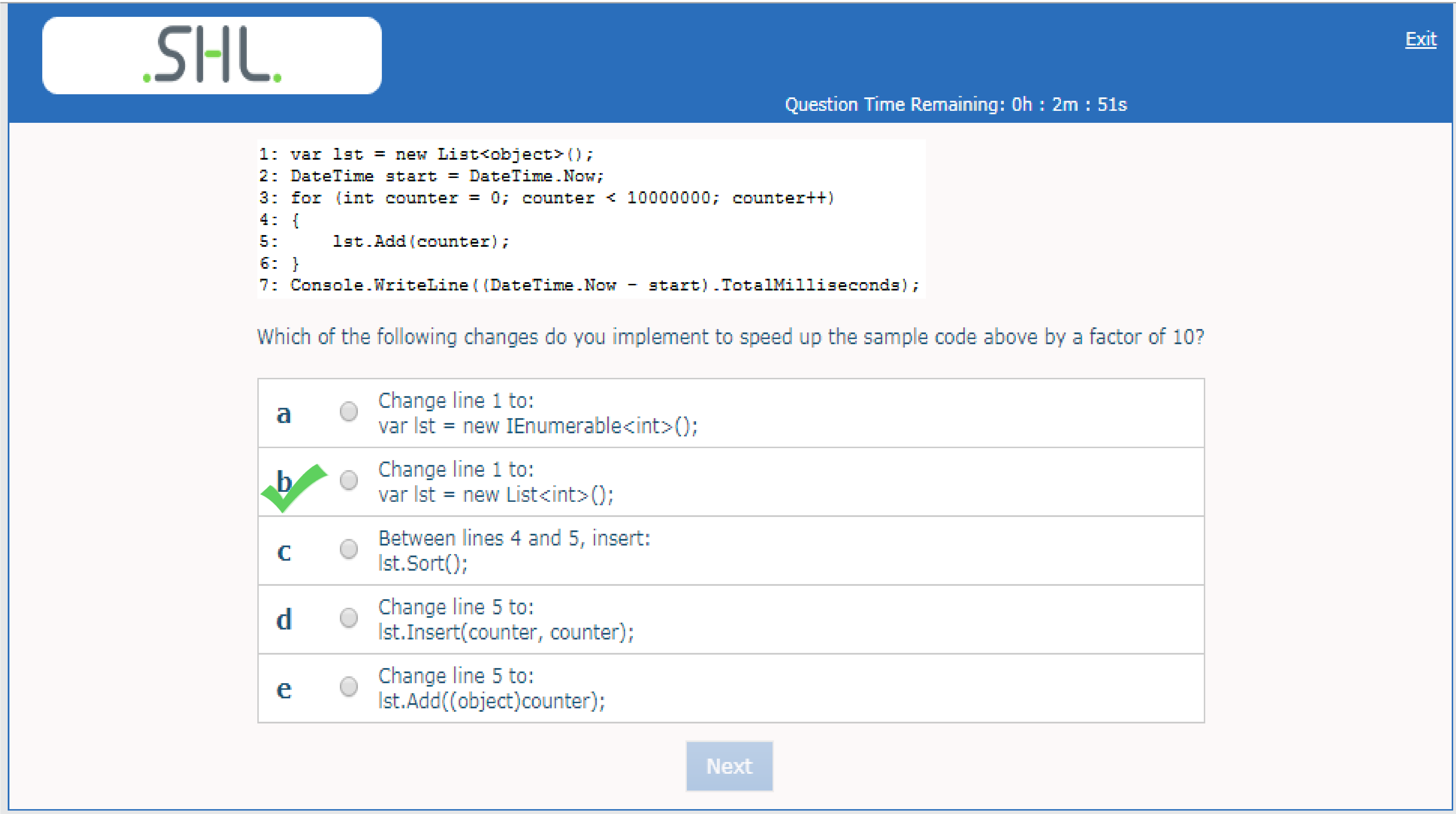
```
interface IDrawable
{
   int Display(left, top, zIndex);
}
```

d

```
interface IDrawable
{
   Display(int left, int top, int zIndex) returns void;
}
```

e

```
interface IDrawable
{
   void Display[int left, int top, int zIndex];
}
```



```
1: var lst = new List<object>();
2: DateTime start = DateTime.Now;
3: for (int counter = 0; counter < 10000000; counter++)
4: {
5:     lst.Add(counter);
6: }
7: Console.WriteLine((DateTime.Now - start).TotalMilliseconds);
```

Which of the following changes do you implement to speed up the sample code above by a factor of 10?

a. Change line 1 to:
var lst = new IEnumerable<int>();

[✓ b.] Change line 1 to:
var lst = new List<int>();

c. Between lines 4 and 5, insert:
lst.Sort();

d. Change line 5 to:
lst.Insert(counter, counter);

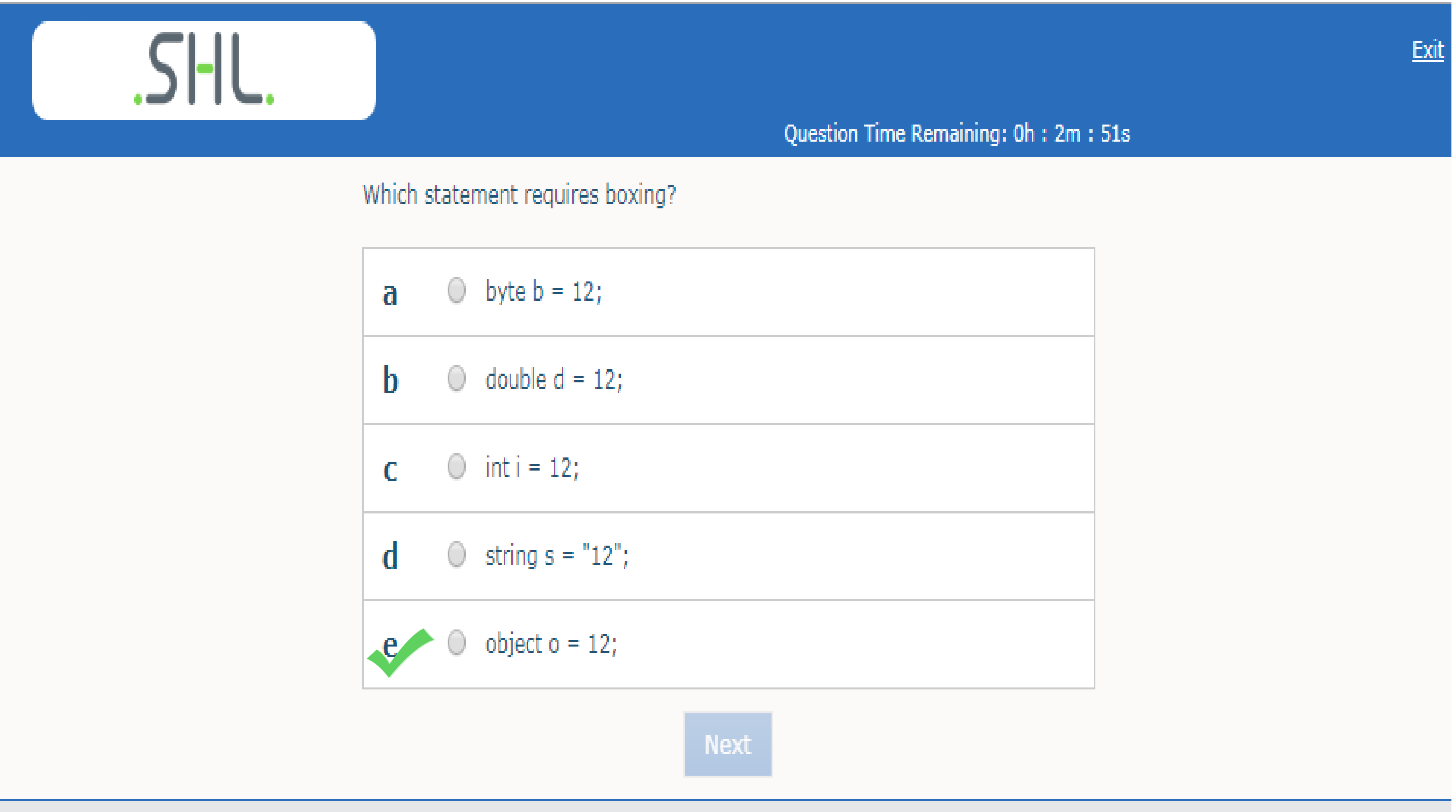
e. Change line 5 to:
lst.Add((object)counter);

```
static void Main()
   {
       var stack = new Stack<int>();
       var queue = new Queue<int>();
       foreach (var i in new int[] { 1, 2})
       {
           stack.Push(i);
           queue.Enqueue(i);
       }
       foreach (var i in stack)
       {
           Console.WriteLine(i);
       }
       foreach (var i in queue)
       {
           Console.WriteLine(i);
       }

   }
```

What is written to the console after the sample code above executes?

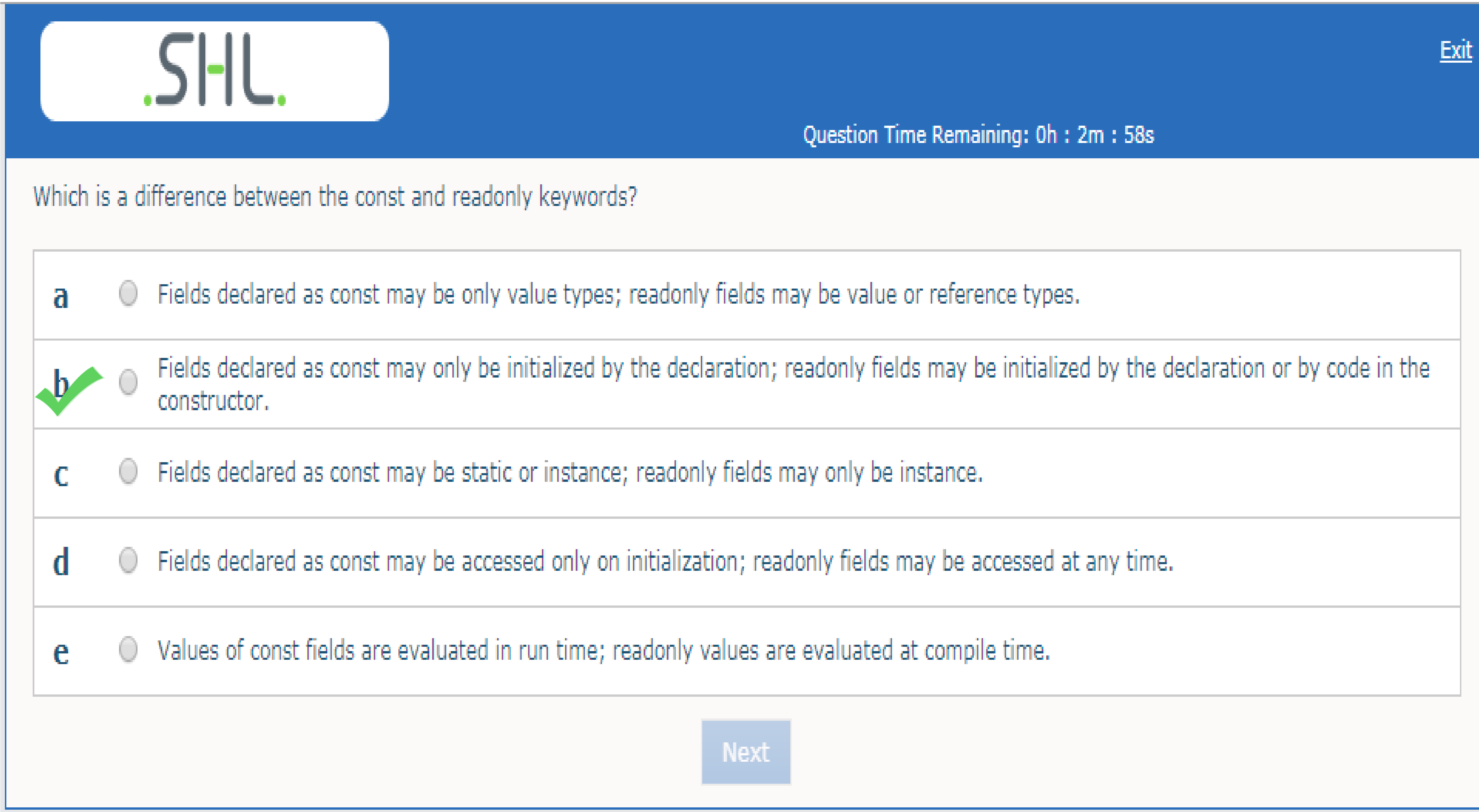
a	○	1 1 2 2
b	○	1 2 1 2
[✓ c]	○	2 1 1 2
d	○	2 1 2 1
e	○	2 2 1 1

Next



Which statement requires boxing?

a	byte b = 12;
b	double d = 12;
c	int i = 12;
d	string s = "12";
✓ e	object o = 12;



Exit

Question Time Remaining: 0h : 2m : 58s

Which is a difference between the const and readonly keywords?

a ○ Fields declared as const may be only value types; readonly fields may be value or reference types.

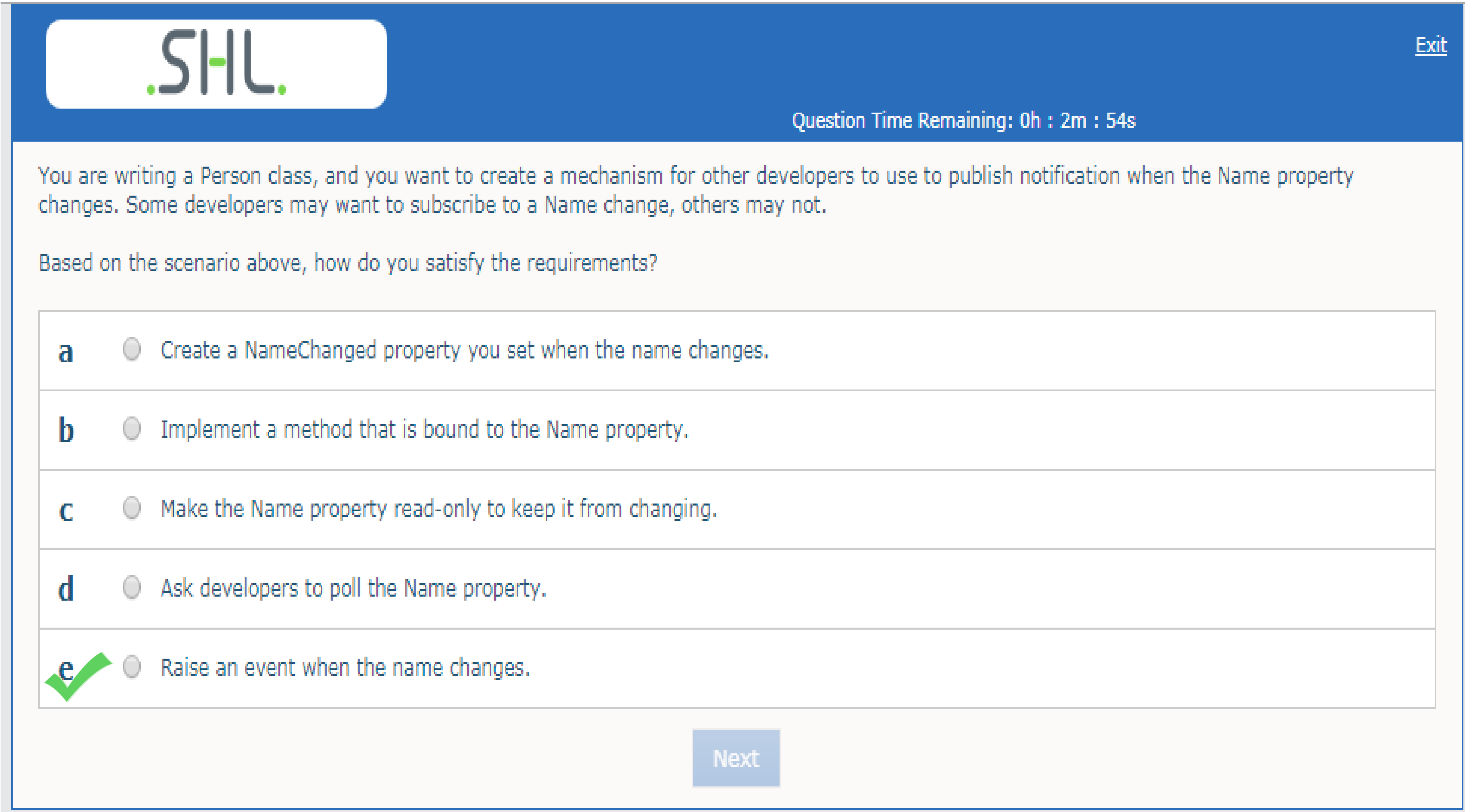
[✓ b] ○ Fields declared as const may only be initialized by the declaration; readonly fields may be initialized by the declaration or by code in the constructor.

c ○ Fields declared as const may be static or instance; readonly fields may only be instance.

d ○ Fields declared as const may be accessed only on initialization; readonly fields may be accessed at any time.

e ○ Values of const fields are evaluated in run time; readonly values are evaluated at compile time.

Next



Exit

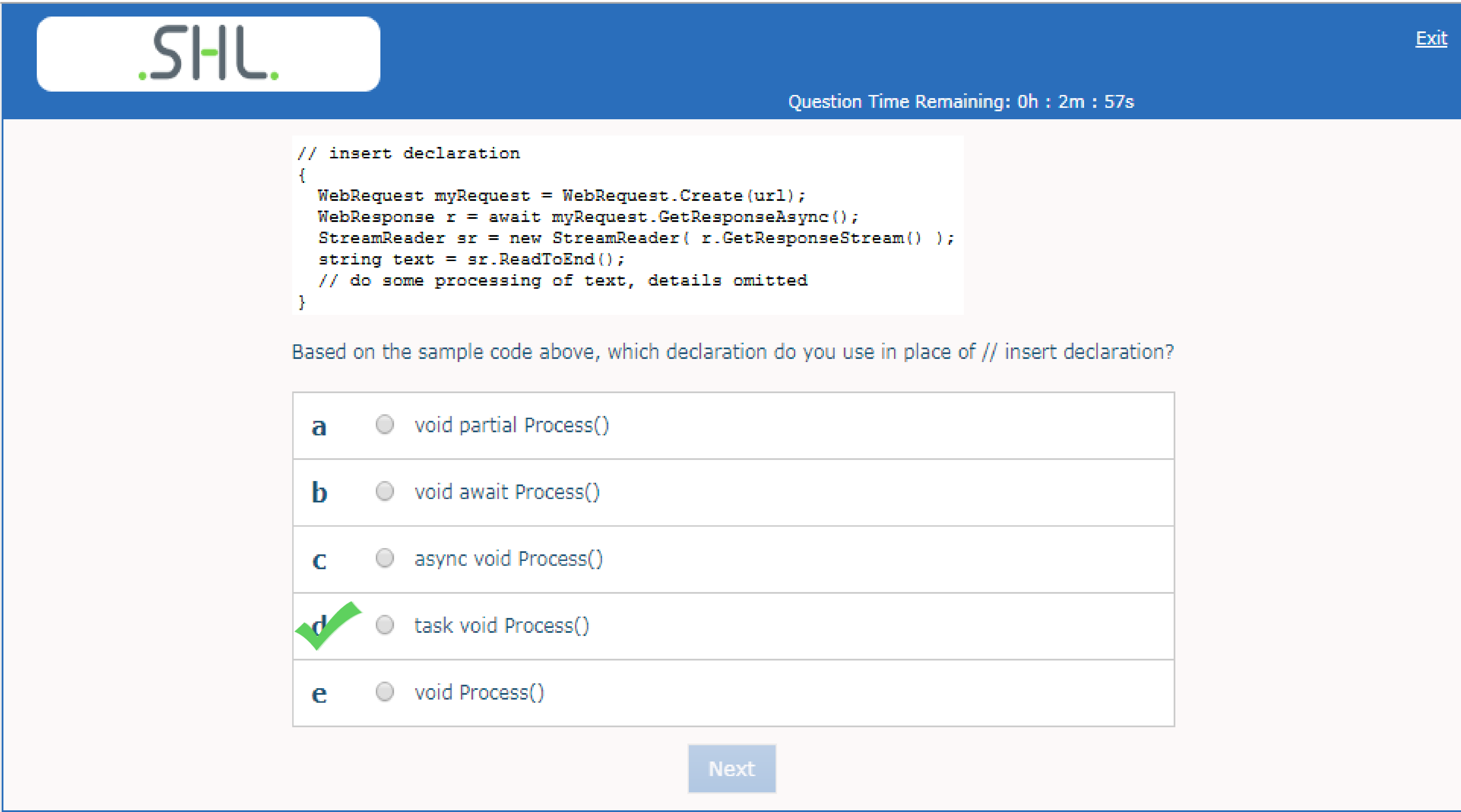
Question Time Remaining: 0h : 2m : 54s

You are writing a Person class, and you want to create a mechanism for other developers to use to publish notification when the Name property changes. Some developers may want to subscribe to a Name change, others may not.

Based on the scenario above, how do you satisfy the requirements?

a	Create a NameChanged property you set when the name changes.
b	Implement a method that is bound to the Name property.
c	Make the Name property read-only to keep it from changing.
d	Ask developers to poll the Name property.
[✓ e]	Raise an event when the name changes.

Next



```
// insert declaration
{
  WebRequest myRequest = WebRequest.Create(url);
  WebResponse r = await myRequest.GetResponseAsync();
  StreamReader sr = new StreamReader( r.GetResponseStream() );
  string text = sr.ReadToEnd();
  // do some processing of text, details omitted
}
```

Based on the sample code above, which declaration do you use in place of // insert declaration?

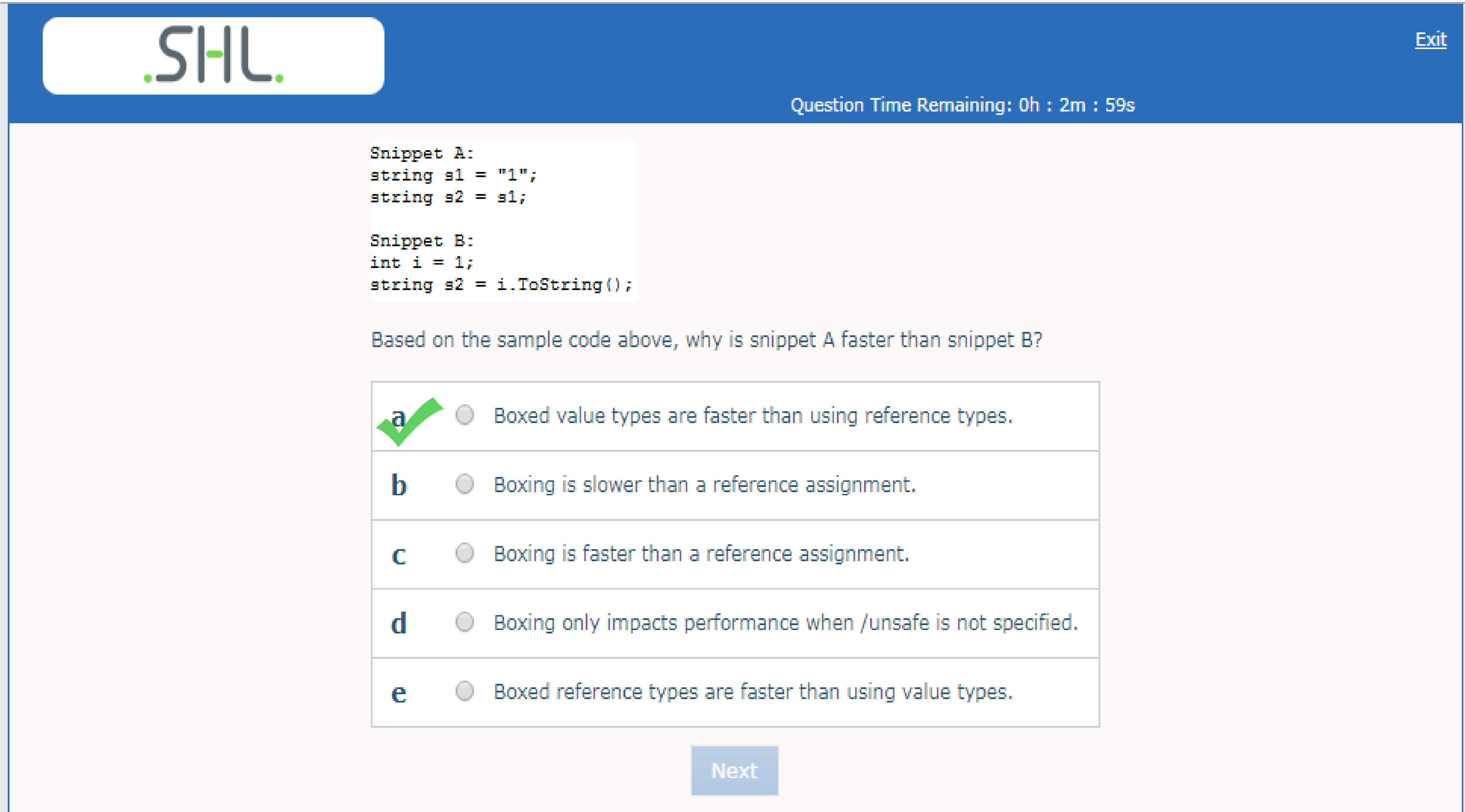
a void partial Process()

b void await Process()

c async void Process()

[✓ d] task void Process()

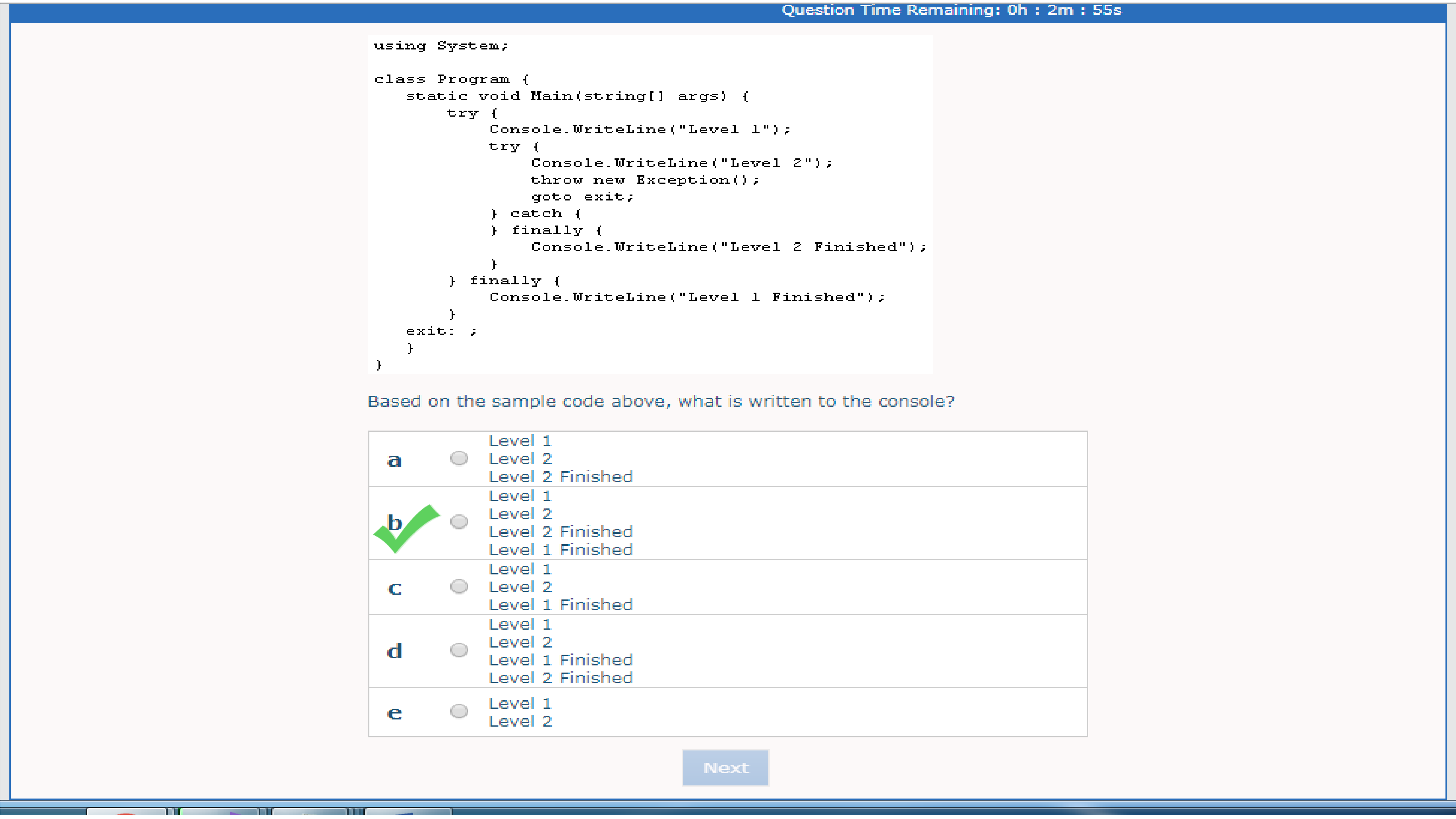
e void Process()



```
Snippet A:
string s1 = "1";
string s2 = s1;

Snippet B:
int i = 1;
string s2 = i.ToString();
```

Based on the sample code above, why is snippet A faster than snippet B?

[✓ a]	Boxed value types are faster than using reference types.
b	Boxing is slower than a reference assignment.
c	Boxing is faster than a reference assignment.
d	Boxing only impacts performance when /unsafe is not specified.
e	Boxed reference types are faster than using value types.



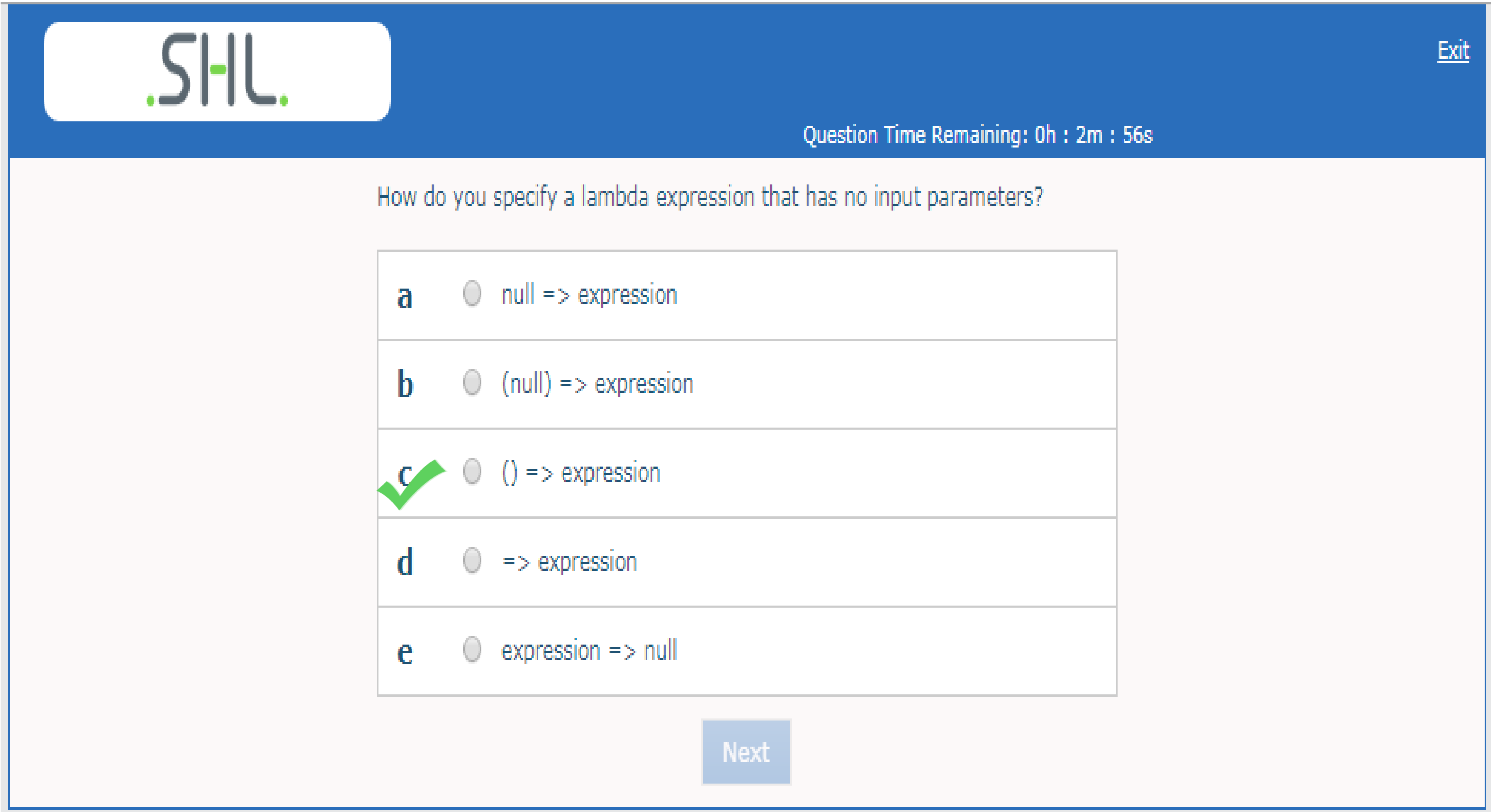
Question Time Remaining: 0h : 2m : 55s

```
using System;

class Program {
   static void Main(string[] args) {
       try {
           Console.WriteLine("Level 1");
           try {
               Console.WriteLine("Level 2");
               throw new Exception();
               goto exit;
           } catch {
           } finally {
               Console.WriteLine("Level 2 Finished");
           }
       } finally {
           Console.WriteLine("Level 1 Finished");
       }
   exit: ;
   }
}
```

Based on the sample code above, what is written to the console?

a	○	Level 1 Level 2 Level 2 Finished
✓ b	○	Level 1 Level 2 Level 2 Finished Level 1 Finished
c	○	Level 1 Level 2 Level 1 Finished
d	○	Level 1 Level 2 Level 1 Finished Level 2 Finished
e	○	Level 1 Level 2

Next



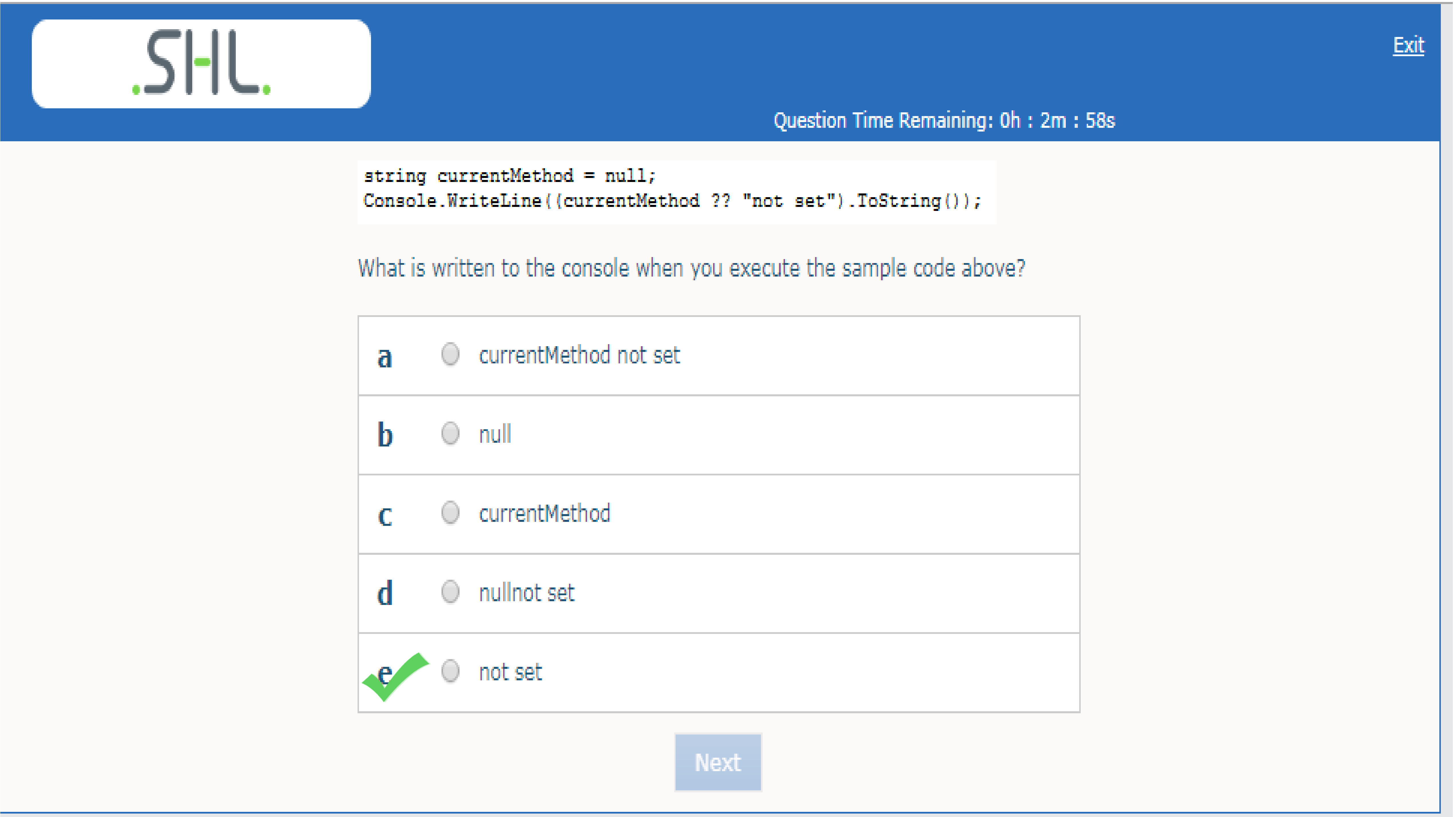
Exit

Question Time Remaining: 0h : 2m : 56s

How do you specify a lambda expression that has no input parameters?

a	null => expression
b	(null) => expression
[✓ c]	() => expression
d	=> expression
e	expression => null

Next



```
string currentMethod = null;
Console.WriteLine((currentMethod ?? "not set").ToString());
```

What is written to the console when you execute the sample code above?

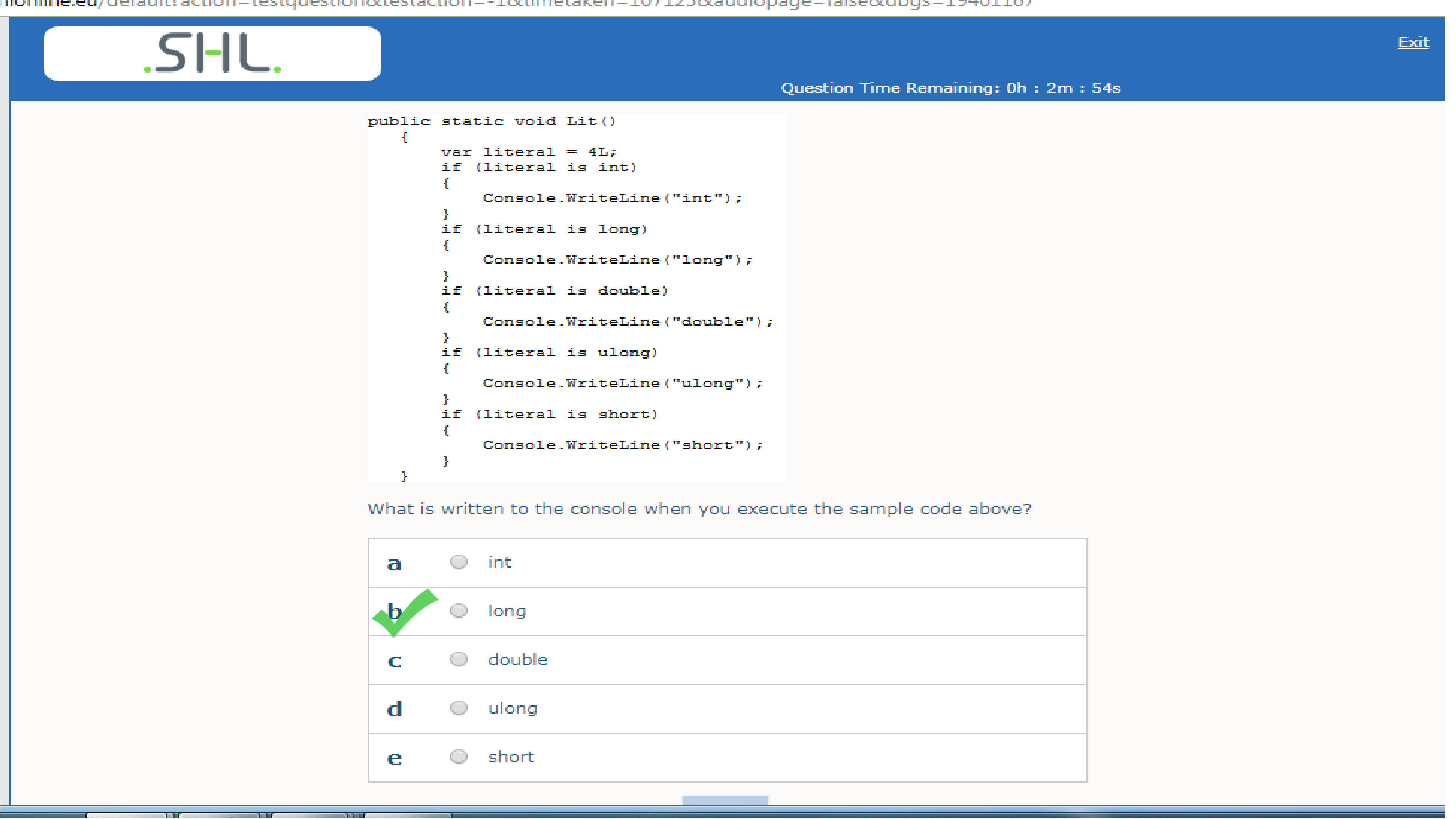
a ○ currentMethod not set

b ○ null

c ○ currentMethod

d ○ nullnot set

[✓ e] ○ not set



lonline.eu/default?action=testquestion&testaction=-1&timetaken=107123&audiopage=false&dbgs=19401167

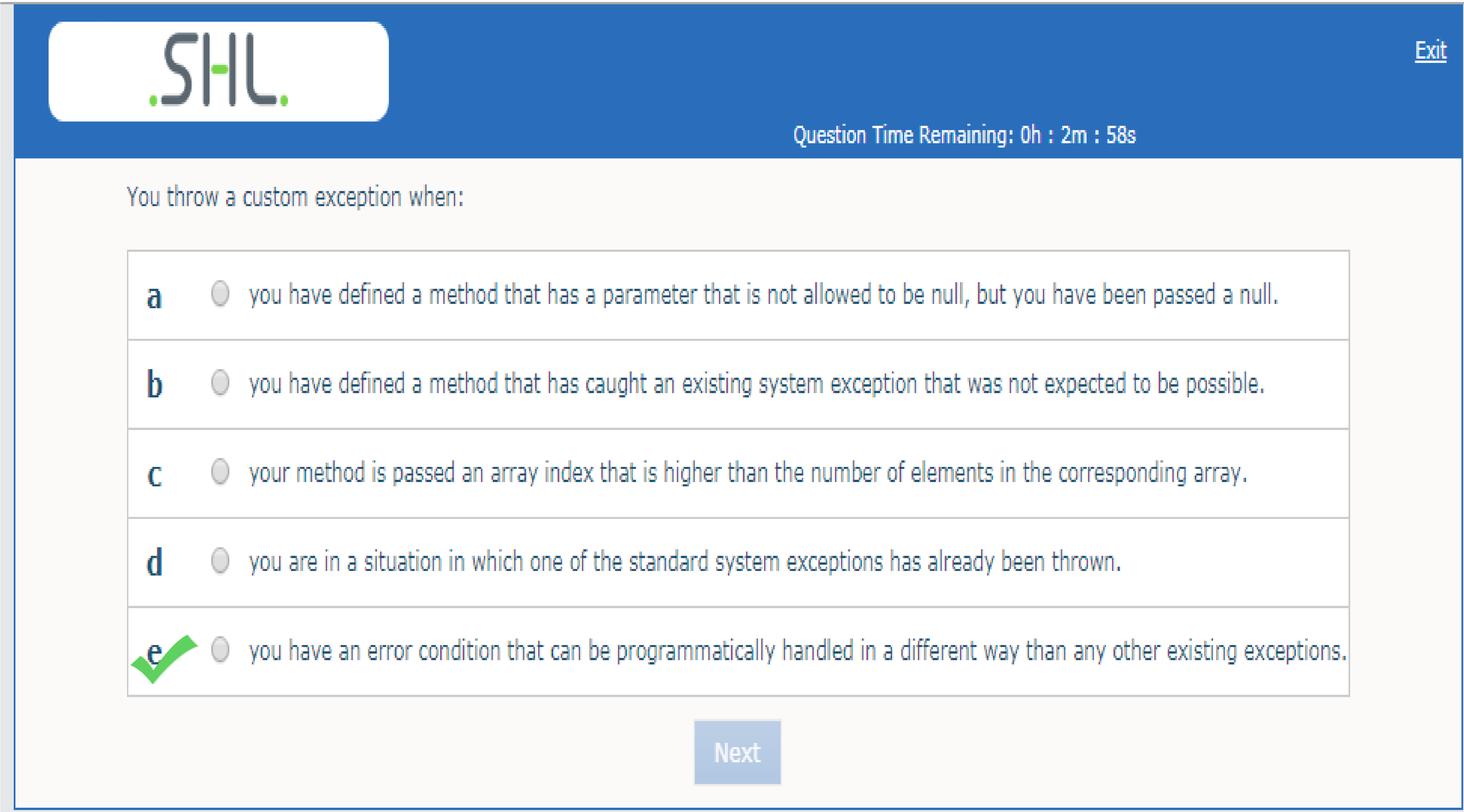
Exit

Question Time Remaining: 0h : 2m : 54s

```
public static void Lit()
    {
        var literal = 4L;
        if (literal is int)
        {
            Console.WriteLine("int");
        }
        if (literal is long)
        {
            Console.WriteLine("long");
        }
        if (literal is double)
        {
            Console.WriteLine("double");
        }
        if (literal is ulong)
        {
            Console.WriteLine("ulong");
        }
        if (literal is short)
        {
            Console.WriteLine("short");
        }
    }
```

What is written to the console when you execute the sample code above?

a	○	int
[✓ b]	○	long
c	○	double
d	○	ulong
e	○	short



Question Time Remaining: 0h : 2m : 58s

You throw a custom exception when:

a ○ you have defined a method that has a parameter that is not allowed to be null, but you have been passed a null.

b ○ you have defined a method that has caught an existing system exception that was not expected to be possible.

c ○ your method is passed an array index that is higher than the number of elements in the corresponding array.

d ○ you are in a situation in which one of the standard system exceptions has already been thrown.

[✓ e] ○ you have an error condition that can be programmatically handled in a different way than any other existing exceptions.

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

✓ a ○
```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

b ○
```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

c ○
```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

d ○
```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

e ○
```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
public static IEnumerable ParseCodes(string value, out int code)
{
    int parsedCode;
    code = 0;

    if (!Int32.TryParse(value, out parsedCode)) {
        yield return "false";
    }

    code = parsedCode;

    for (int i = 0; i <= parsedCode.ToString( ).Length; i++) {
        yield return parsedCode.ToString( ).Substring(i, 1);
    }
}

static void Main(string[] args) {
    int code;
    foreach (string n in ParseCodes("[82738]", out code)) {
        Console.WriteLine(n);
    }

    Console.WriteLine("Code: " + code);
    Console.ReadLine( );
}
```

Why is there an error when you compile the sample code above?

a	○	Main must be declared public.
✓ b	○	You cannot use ref/out parameters with an iterator.
c	○	You cannot assign a value to an out parameter more than once.
d	○	TryParse is not a valid method.
e	○	Return cannot follow yield.

```
static void Main()
{
    Compound x = new Compound();
    x.Mix(ref x.NumberOfTimesMixed);
    Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
    public int NumberOfTimesMixed { get; set;}

    public void Mix(ref int counter, params Compound[] components)
    {
         counter++;
         // further processing
    }
    }
```

How do you fix the sample code above?

a ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
x.Mix(x.NumberOfTimesMixed);
```

b ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
x.Mix(ref x.NumberOfTimesMixed, x);
```

c ○ Insert

```
public Compound() {}
```

above

```
public int NumberOfTimesMixed { get; set;}
```

✓ d ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
int i = x.NumberOfTimesMixed;
x.Mix(ref i);
x.NumberOfTimesMixed = i;
```

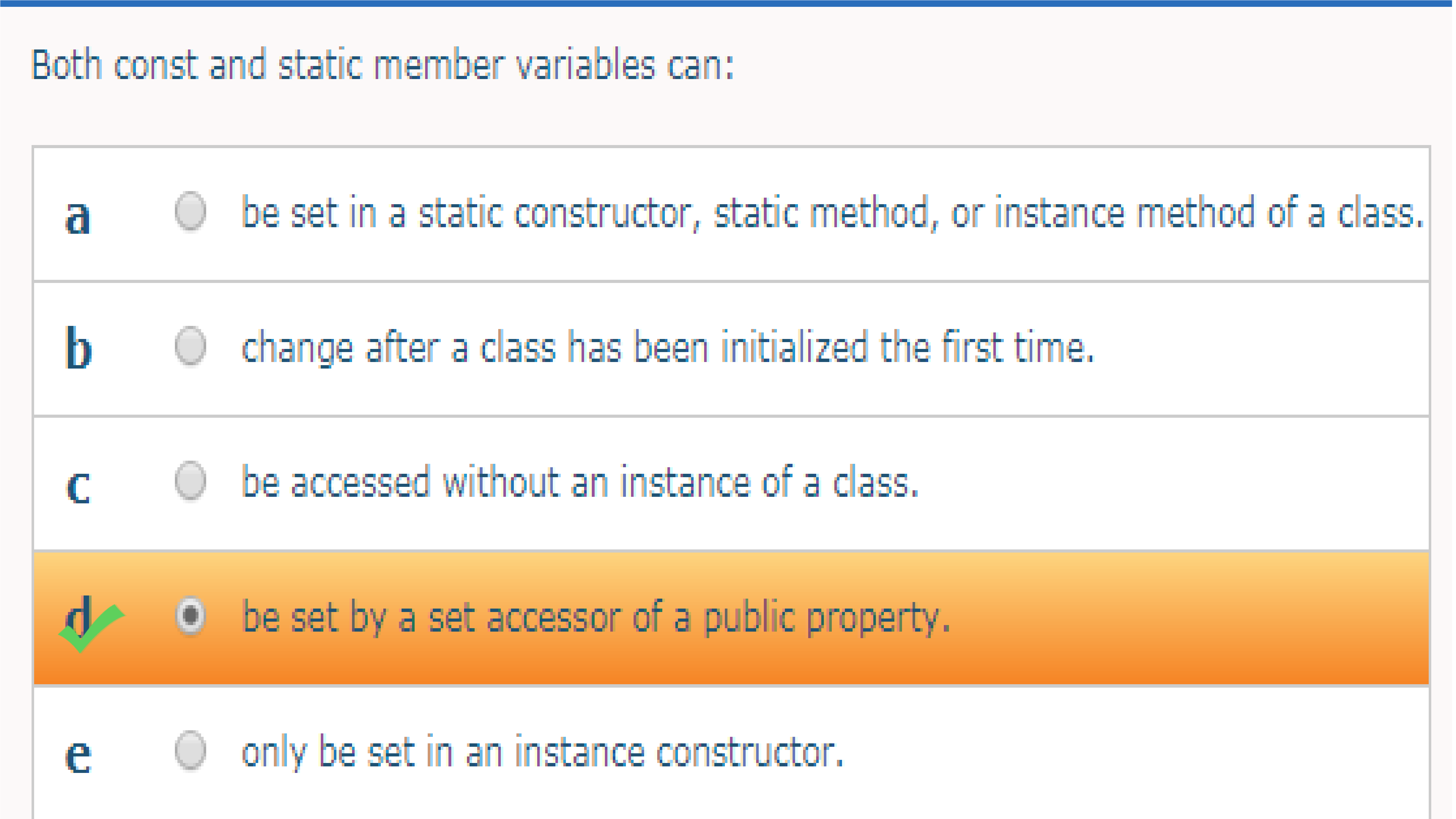
e ○ Remove

```
public int NumberOfTimesMixed { get; set;}
```

```
static object StartList(object[] array)
    {
        foreach (var o in array)
        {
            if (o is IEnumerable)
            {
                return ((IEnumerable)o).GetEnumerator();
            }
        }
        return Enumerable.Range(0, 4).GetEnumerator();
    }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	○ ICollection
b	○ IEnumerable
c	○ IList
✓ d	○ IEnumerator
e	○ IArray



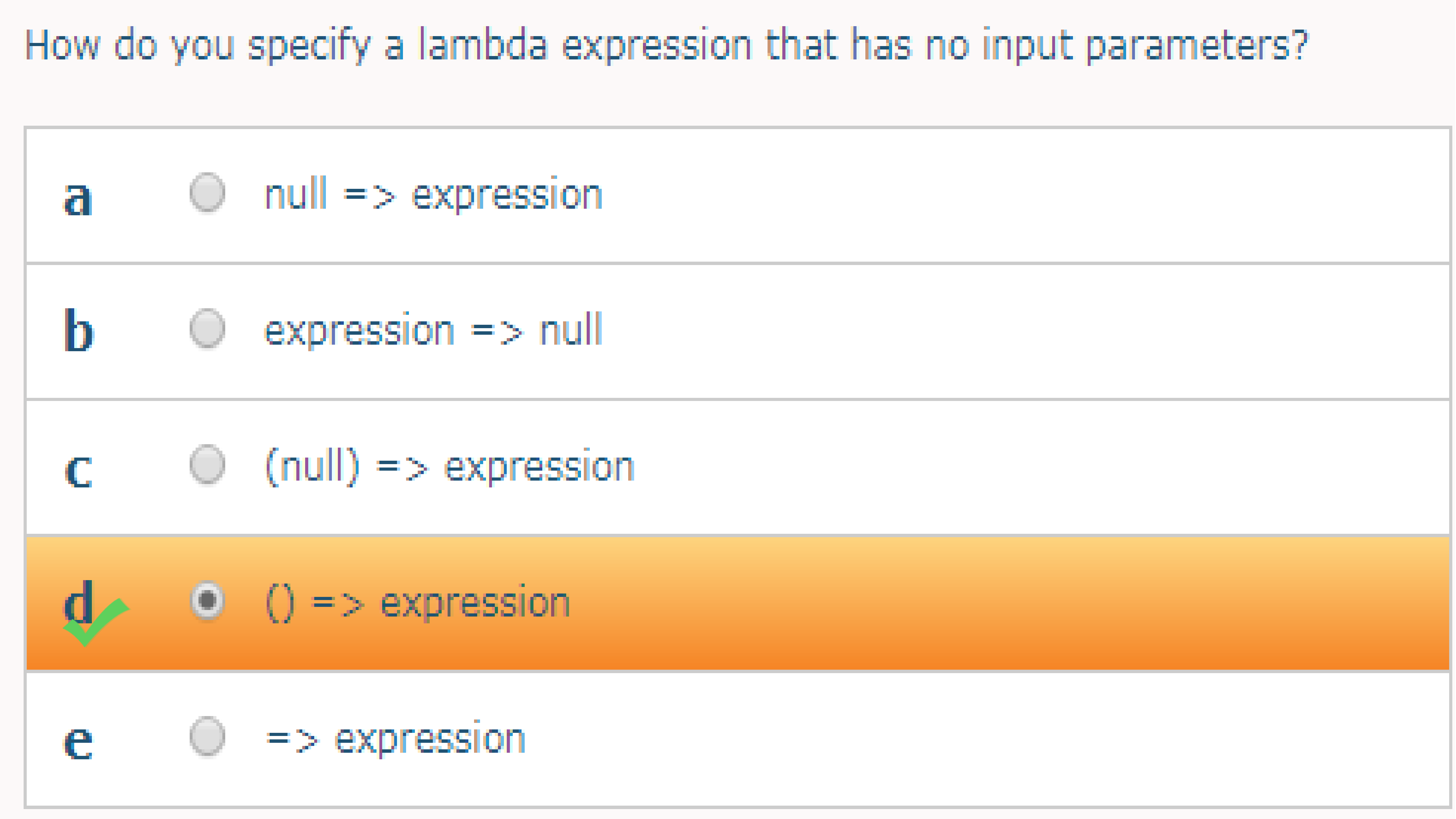
Both const and static member variables can:

- a ○ be set in a static constructor, static method, or instance method of a class.
- b ○ change after a class has been initialized the first time.
- c ○ be accessed without an instance of a class.
- [✓ d ● be set by a set accessor of a public property.]
- e ○ only be set in an instance constructor.

You have a class "Person" that has a method named "GetIdentifier". You call this method in the class definition, and you want derived classes such as "Employee" or "BoardMember" to be able to call this method, even if these classes are derived in code added by your customer base. However, you do not want to allow this method to be called anywhere else.

In order to achieve the objectives in the scenario above, you declare the GetIdentifier method as:

a	○	protected in Person.
b	○	private in derived classes.
[✓ c]	○	private in Person.
d	○	protected in derived classes.
e	○	internal in Person.



How do you specify a lambda expression that has no input parameters?

- a `null => expression`
- b `expression => null`
- c `(null) => expression`
- [✓ d `() => expression`]
- e `=> expression`

Which is the default underlying type for the values of an enum?

a	○	char
b	○	short
c	○	byte
d ✓	○	int
e	○	long

Which method do you call to prevent the Finalize method from being called when implementing a Dispose() method?

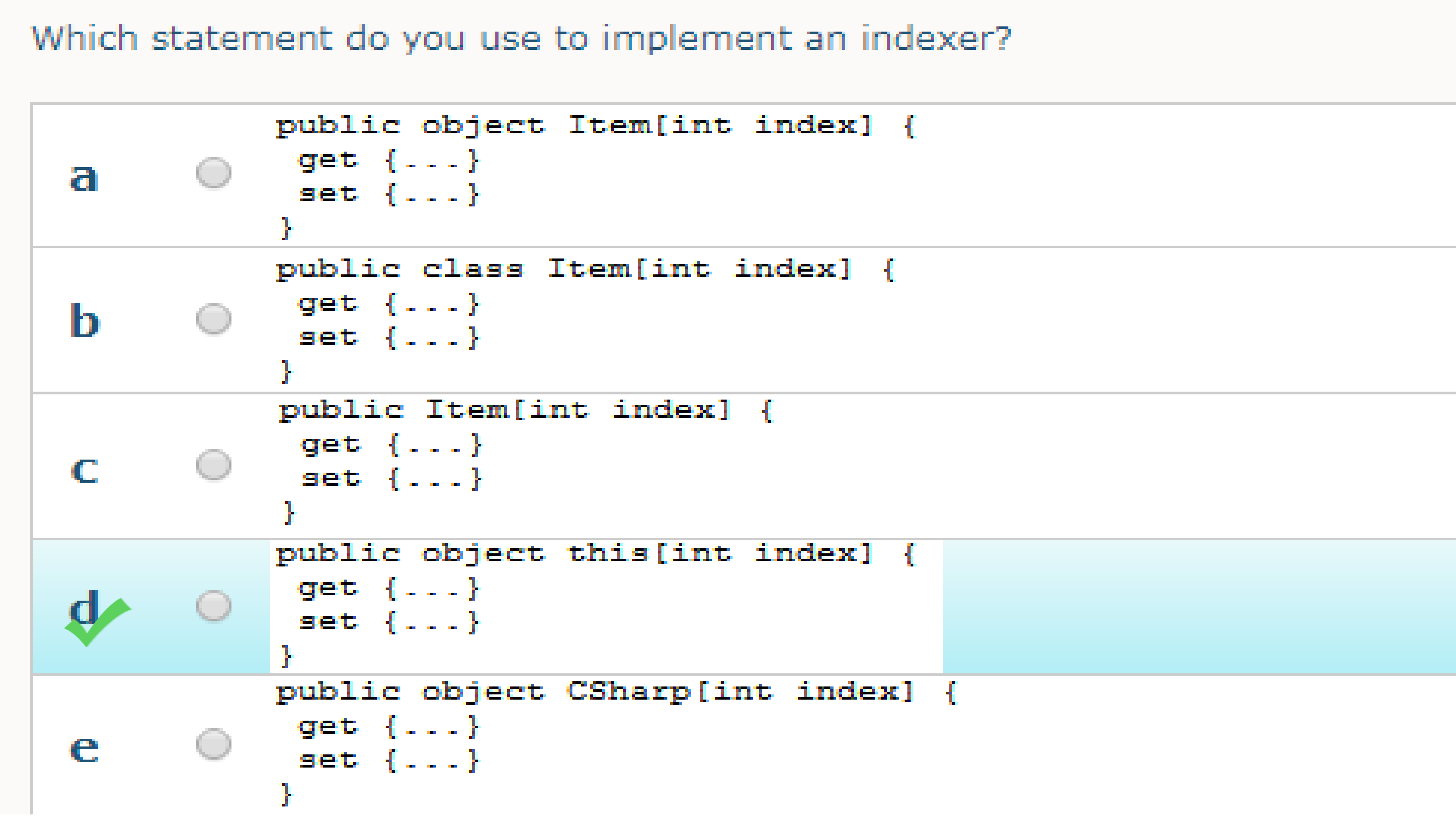
a ○ GC.SuppressFinalization

b ○ GC.NoFinalize

[✓ c] ○ GC.SuppressFinalize

d ○ System.NoFinalization

e ○ GC.NoFinalization



Which statement do you use to implement an indexer?

a ○
```
public object Item[int index] {
 get {...}
 set {...}
}
```

b ○
```
public class Item[int index] {
 get {...}
 set {...}
}
```

c ○
```
public Item[int index] {
 get {...}
 set {...}
}
```

[✓ d] ○
```
public object this[int index] {
 get {...}
 set {...}
}
```

e ○
```
public object CSharp[int index] {
 get {...}
 set {...}
}
```

There is a collection of integers in the variable "col". You want to project those integers in "col" which are divisible by four into the variable "dst".

Based on the scenario above, which statement do you use to accomplish your goal?

a ○ `var dst = from c in col where c % 4 == 0 select c;`

b ○ `var dst = select c in col having c % 4 == 0;`

c ○ `var dst = select c from col having c % 4 == 0;`

d ○ `var dst = select c from col where c % 4 == 0;`

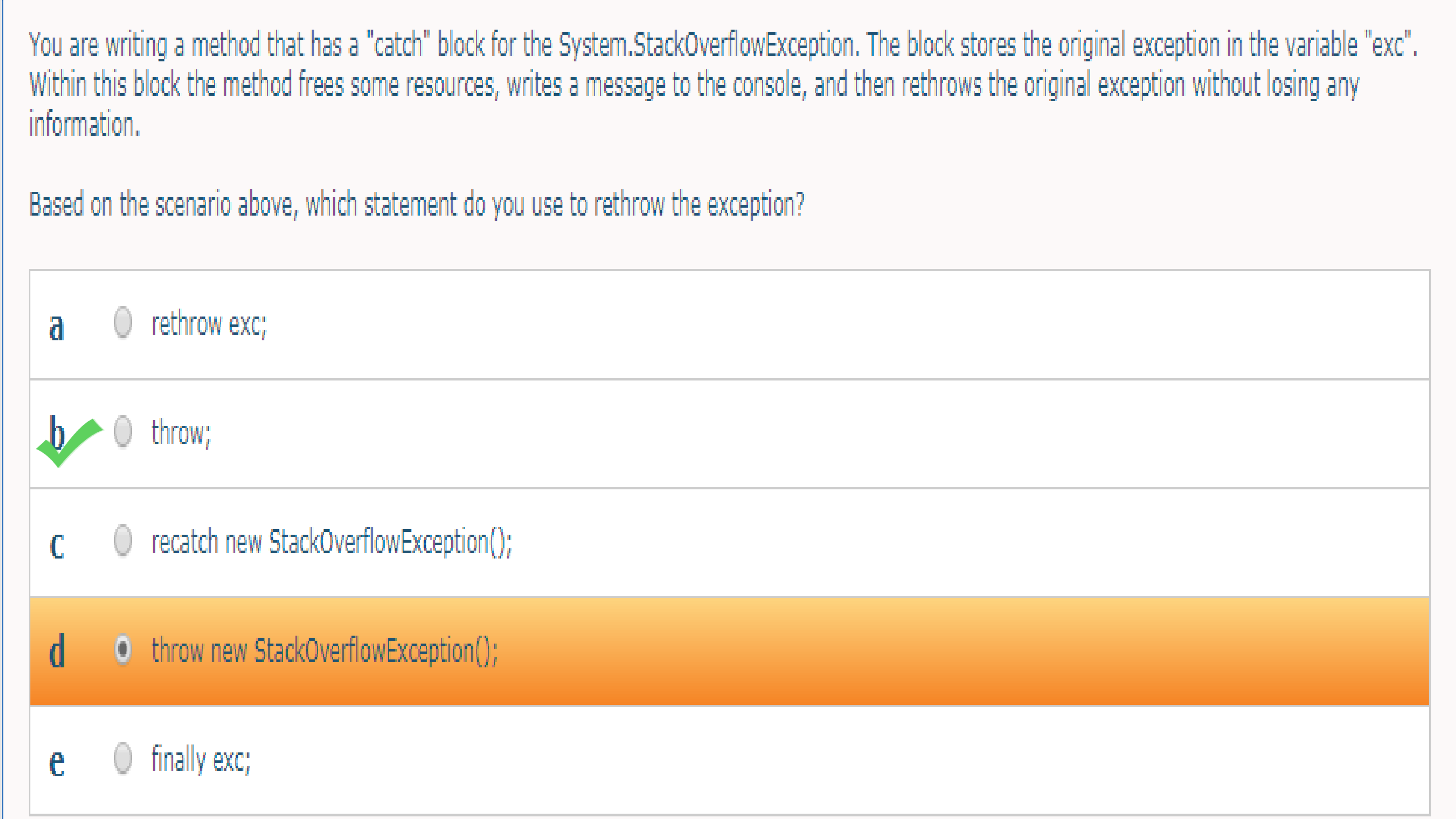
[✓ e] ○ `var dst = where c in col % 4 == 0 select c;`

Which declaration do you use for a method that can be called within the assembly or any derived classes within the same assembly?

a	○	public abstract static void Method()
b	○	private internal virtual void Method()
c	○	public internal static void Method()
[✓ d]	○	protected internal override void Method()
e	○	public override static void Method()

Which is a valid statement?

a	`List<string> objects = new IEnumerable<object>();`
[✓ b]	`IEnumerable<object> objects = new List<string>();`
c	`IEnumerable<string> objects = new List<object>();`
d	`List<object> objects = new IEnumerable<string>();`
e	`IEnumerable<string> objects = new IEnumerable<object>();`



You are writing a method that has a "catch" block for the System.StackOverflowException. The block stores the original exception in the variable "exc". Within this block the method frees some resources, writes a message to the console, and then rethrows the original exception without losing any information.

Based on the scenario above, which statement do you use to rethrow the exception?

a	○	rethrow exc;
b ✓	○	throw;
c	○	recatch new StackOverflowException();
d	●	throw new StackOverflowException();
e	○	finally exc;

```
public void RemoveJim(List<string> Names) {
 foreach(var name in Names) {
   if(name == "Jim") {
     Names.Remove("Jim");
   }
 }
}
```

Based on the sample code above, why is an exception raised when the name Jim exists in the Names collection?

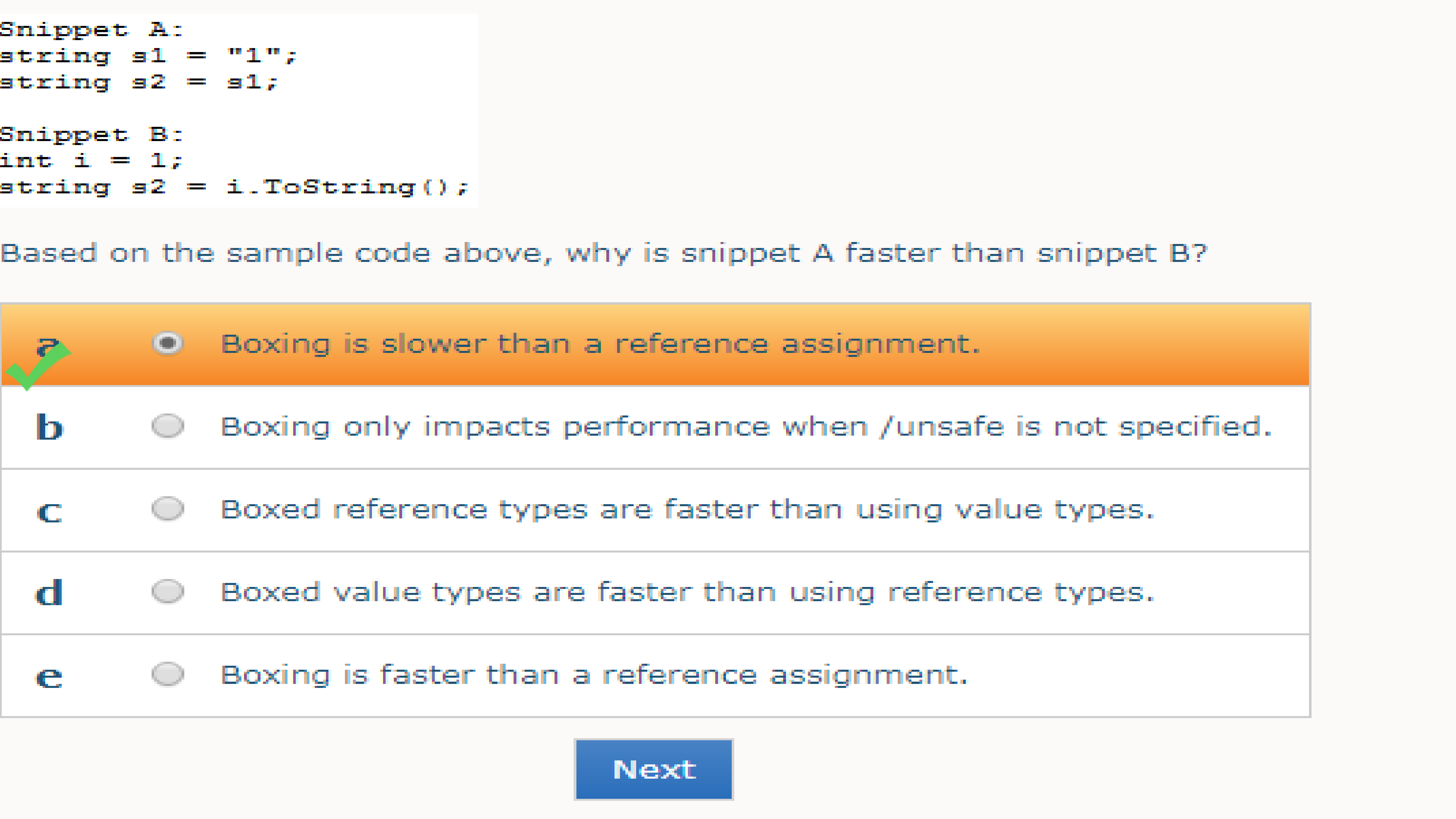
a ✓	○	You cannot modify a collection passed as a parameter to a method.
b	○	Remove does not exist as a member of List<T>.
c	○	The Remove method only accepts an integer.
d	○	You cannot modify a collection while enumerating.
e	○	List is a read-only collection and cannot be altered.

You are creating an Ingredient class, and you want to provide developers a way to combine two Ingredients, forming a new instance of Ingredient. You want to be able to write:

```
public Ingredient Mix(Ingredient a, Ingredient b)
{
  return a + b;
}
```

Based on the scenario above, how do you satisfy the requirements?

a	○	Create a new class called NewIngredient with a constructor that accepts two Ingredients.
b	○	Change the Ingredient class to a struct.
✓ c	○	Overload the + operator in the Ingredient class.
d	○	Implement the IMergeable interface.
e	○	Override the Equals and GetHashCode methods on the Ingredient class.



```
Snippet A:
string s1 = "1";
string s2 = s1;

Snippet B:
int i = 1;
string s2 = i.ToString();
```

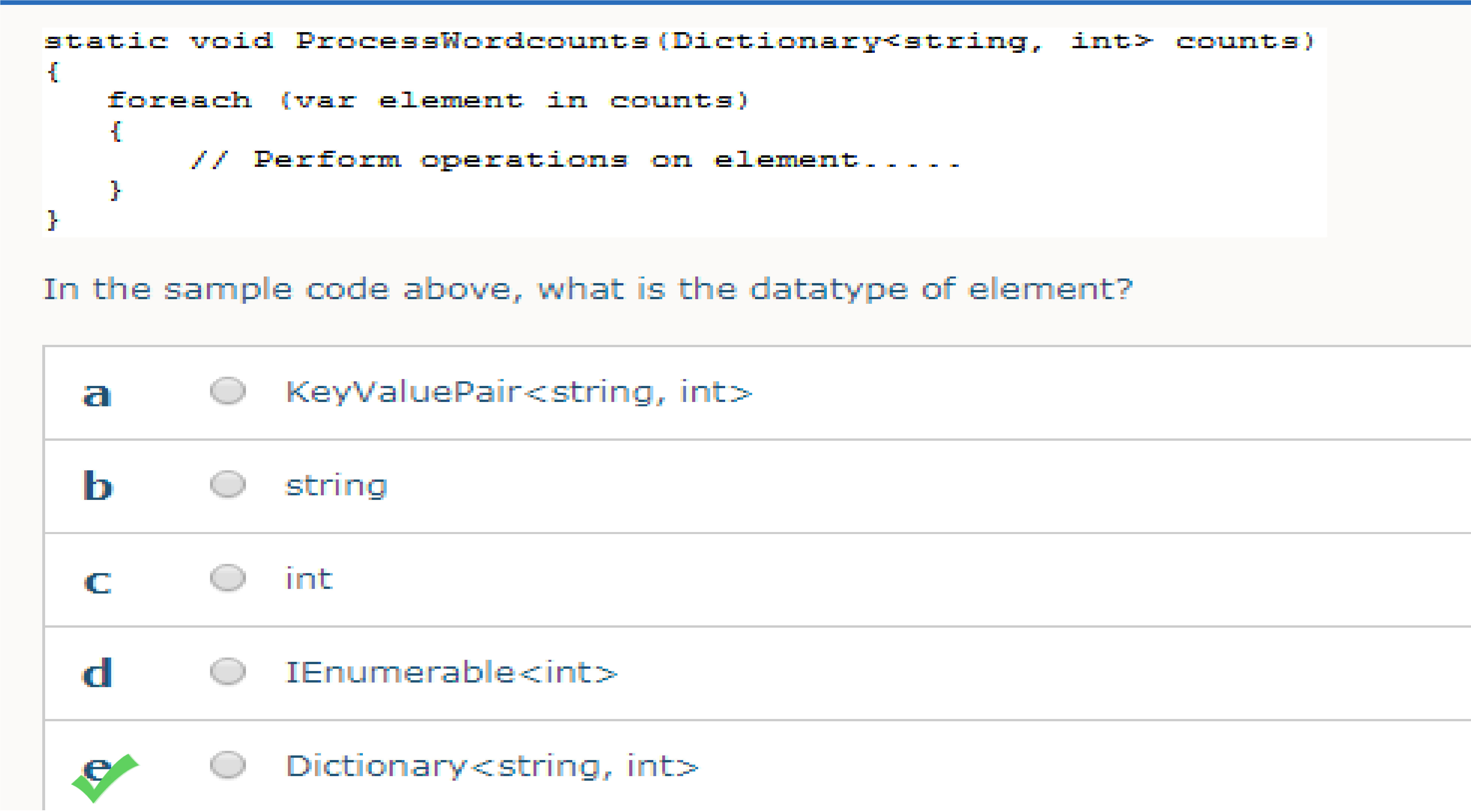
Based on the sample code above, why is snippet A faster than snippet B?

[✓ a]	●	Boxing is slower than a reference assignment.
b	○	Boxing only impacts performance when /unsafe is not specified.
c	○	Boxed reference types are faster than using value types.
d	○	Boxed value types are faster than using reference types.
e	○	Boxing is faster than a reference assignment.

You are writing a compiler and need to keep track of all the identifiers created by the user. Whenever an identifier comes up in the code you are compiling, you need to quickly access the declaration of that identifier via its string value, even if a large number of identifiers are in use.

Based on the scenario above, which data structure do you use to meet the requirements?

a	○	List<string>
[✓ b]	○	Hashtable
c	○	ArrayList
d	○	Queue
e	○	Stack



```
static void ProcessWordcounts(Dictionary<string, int> counts)
{
    foreach (var element in counts)
    {
        // Perform operations on element.....
    }
}
```

In the sample code above, what is the datatype of element?

a	○	KeyValuePair<string, int>
b	○	string
c	○	int
d	○	IEnumerable<int>
e ✓	○	Dictionary<string, int>

```
public class Employee
{
   public virtual void Hire() { /* implement */}
   public virtual void Terminate() { /* implement */}
}

public class Manager : Employee
{
   public override void Hire() { base.Hire(); /* implement more */}
   // implement Terminate
}
```

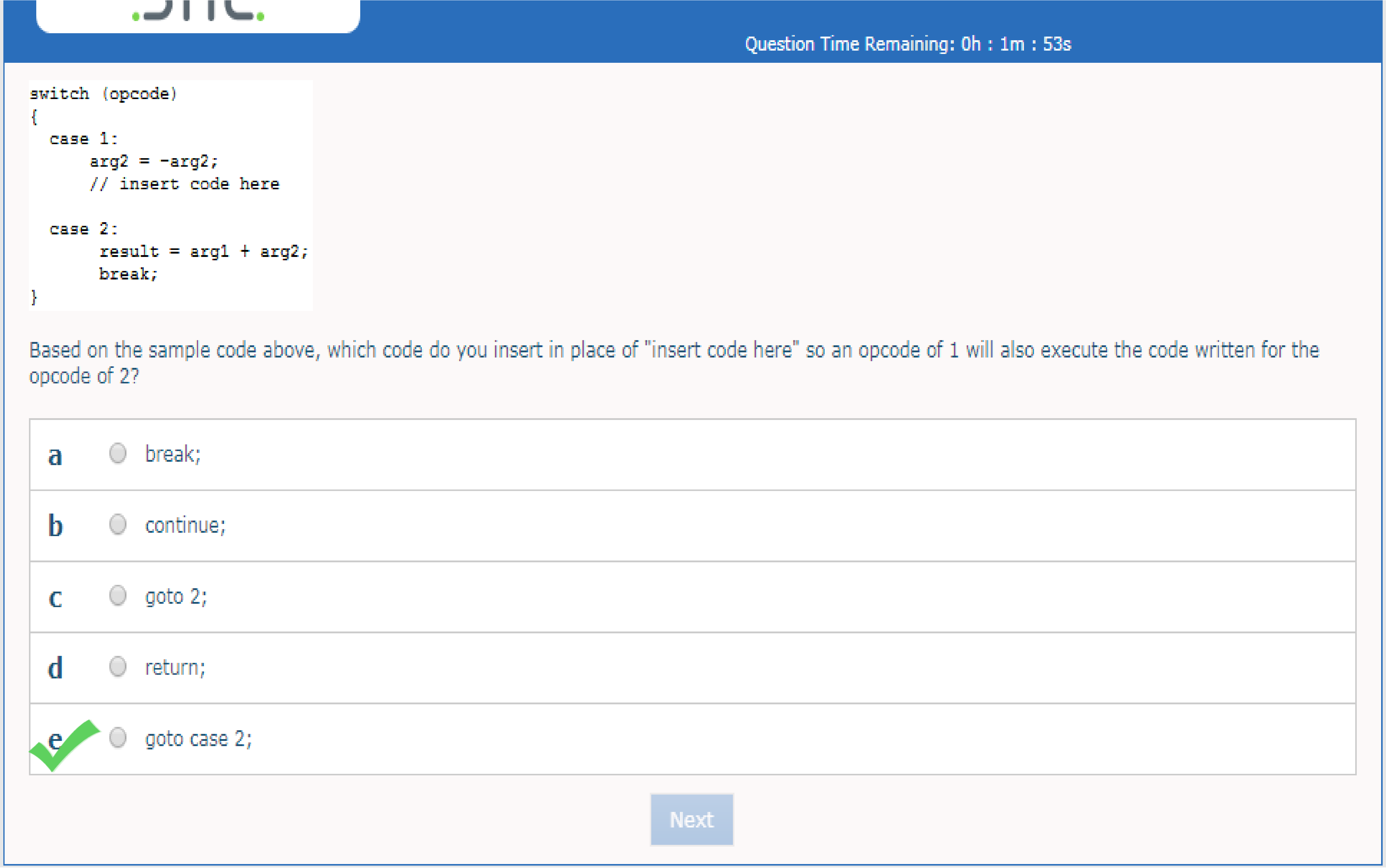
Based on the sample code above, which declaration do you use for Terminate in the class Manager so no class deriving from Manager will override it?

a	○	public virtual void Terminate()
b	○	public volatile void Terminate()
c	○	public static void Terminate()
✓ d	○	public sealed override void Terminate()
e	○	public final override void Terminate()

```
// insert declaration
{
  WebRequest myRequest = WebRequest.Create(url);
  WebResponse r = await myRequest.GetResponseAsync();
  StreamReader sr = new StreamReader( r.GetResponseStream() );
  string text = sr.ReadToEnd();
  // do some processing of text, details omitted
}
```

Based on the sample code above, which declaration do you use in place of // insert declaration?

a	○	void await Process()
b	○	task void Process()
c	○	async void Process()
d	○	void partial Process()
✓ e	○	void Process()



Question Time Remaining: 0h : 1m : 53s

```
switch (opcode)
{
  case 1:
      arg2 = -arg2;
      // insert code here

  case 2:
       result = arg1 + arg2;
       break;
}
```

Based on the sample code above, which code do you insert in place of "insert code here" so an opcode of 1 will also execute the code written for the opcode of 2?

a	break;
b	continue;
c	goto 2;
d	return;
[✓ e]	goto case 2;

Next

```
static void Main() {
   string[] ordinals = new string[] { "First", "Second", "Third",
"Fourth" };
   var taken = ordinals.Take(3);
   Console.WriteLine(taken.Count());
}
```

What is the result of the sample code above?

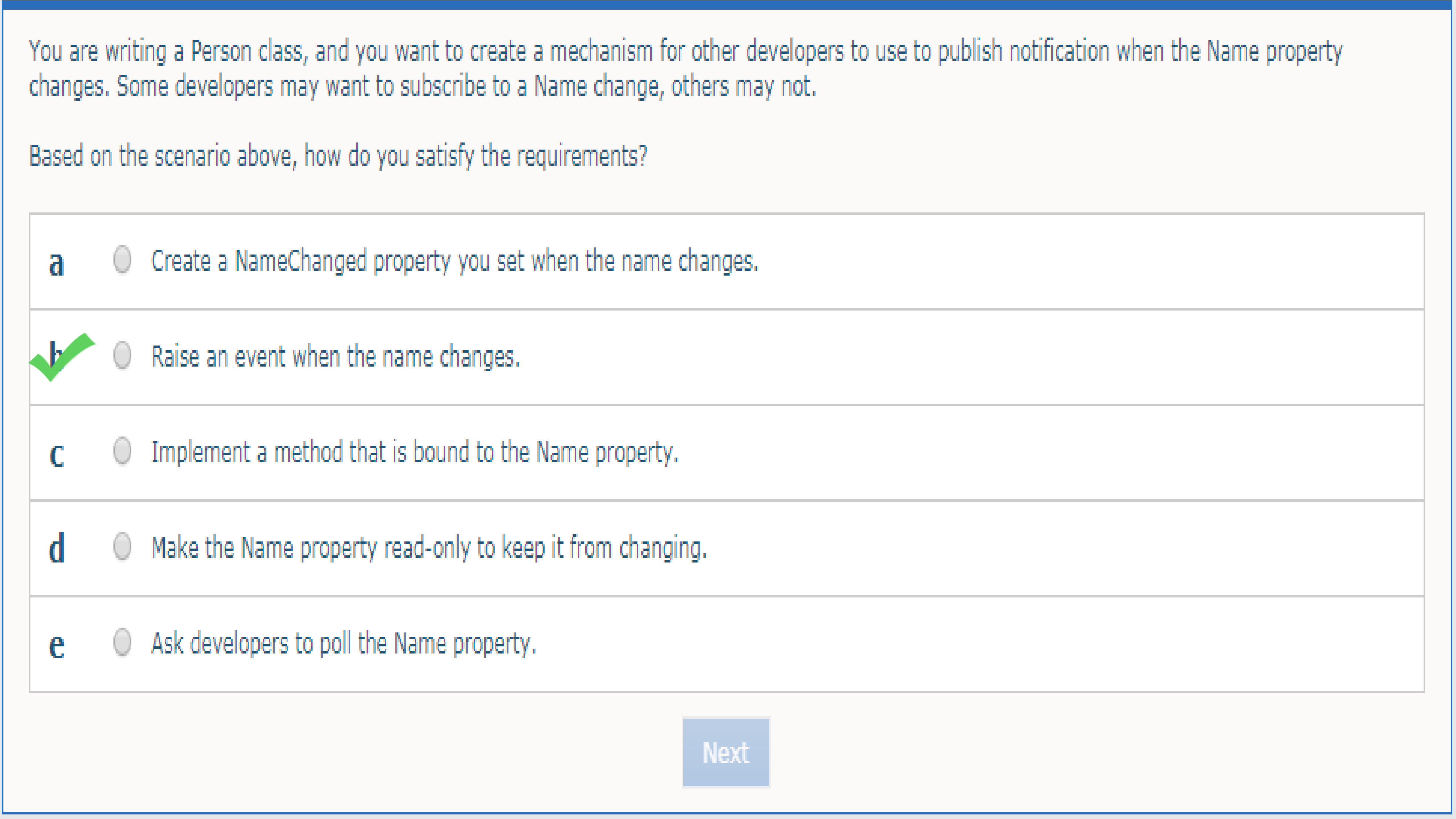
a	A run-time exception is raised on the taken assignment.
b	2 is written to the console.
✓ c	3 is written to the console.
d	4 is written to the console.
e	"Fourth" is written to the console.

Next

```
static void ProcessWordcounts(Dictionary<string, int> counts)
{
   foreach (var element in counts)
   {
       // Perform operations on element.....
   }
}
```

In the sample code above, what is the datatype of element?

a	○ int
b	○ KeyValuePair<string, int>
c ✓	○ Dictionary<string, int>
d	○ string
e	○ IEnumerable<int>

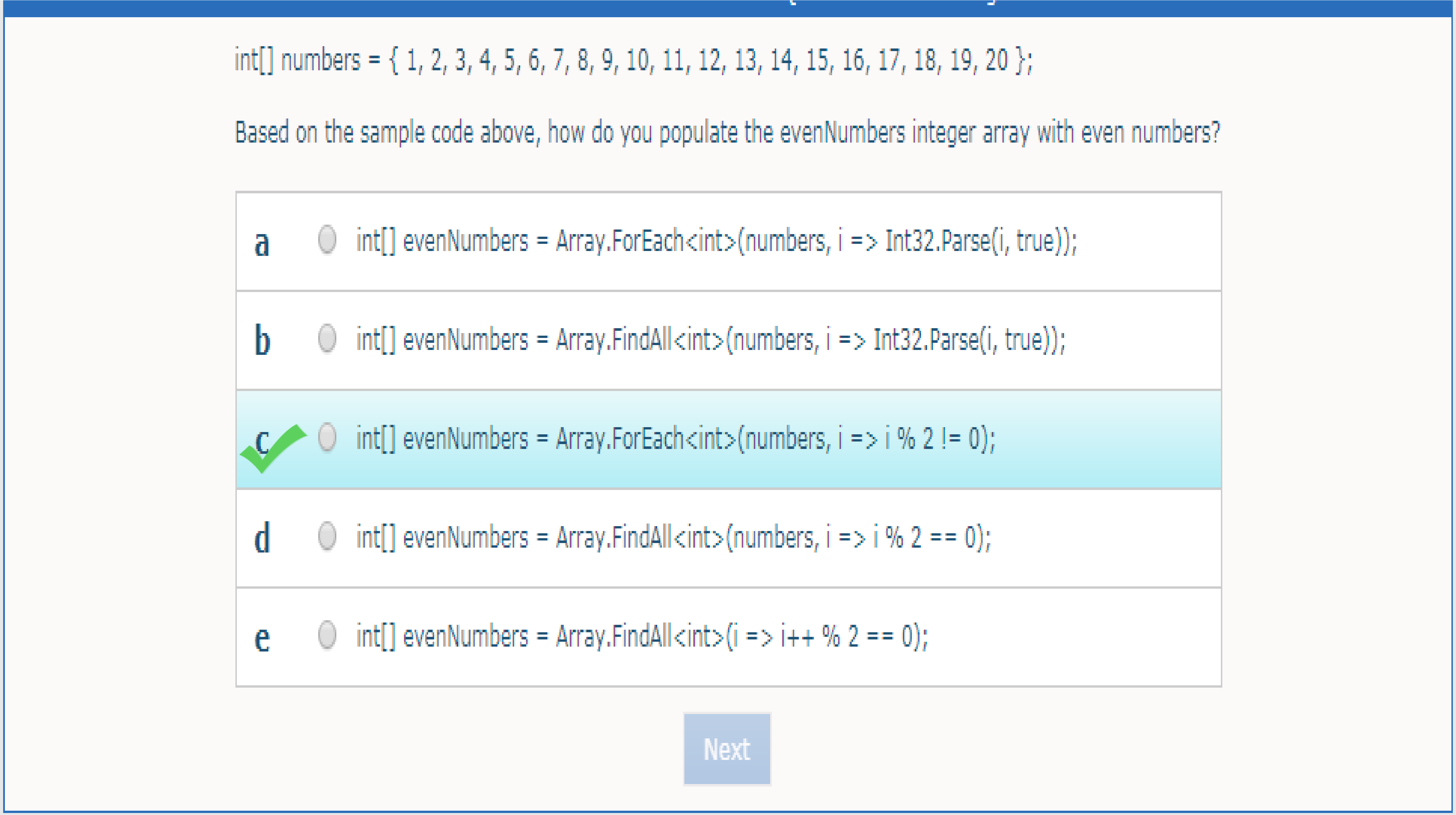


You are writing a Person class, and you want to create a mechanism for other developers to use to publish notification when the Name property changes. Some developers may want to subscribe to a Name change, others may not.

Based on the scenario above, how do you satisfy the requirements?

a	Create a NameChanged property you set when the name changes.
[✓ b]	Raise an event when the name changes.
c	Implement a method that is bound to the Name property.
d	Make the Name property read-only to keep it from changing.
e	Ask developers to poll the Name property.

Next



```
int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };
```

Based on the sample code above, how do you populate the evenNumbers integer array with even numbers?

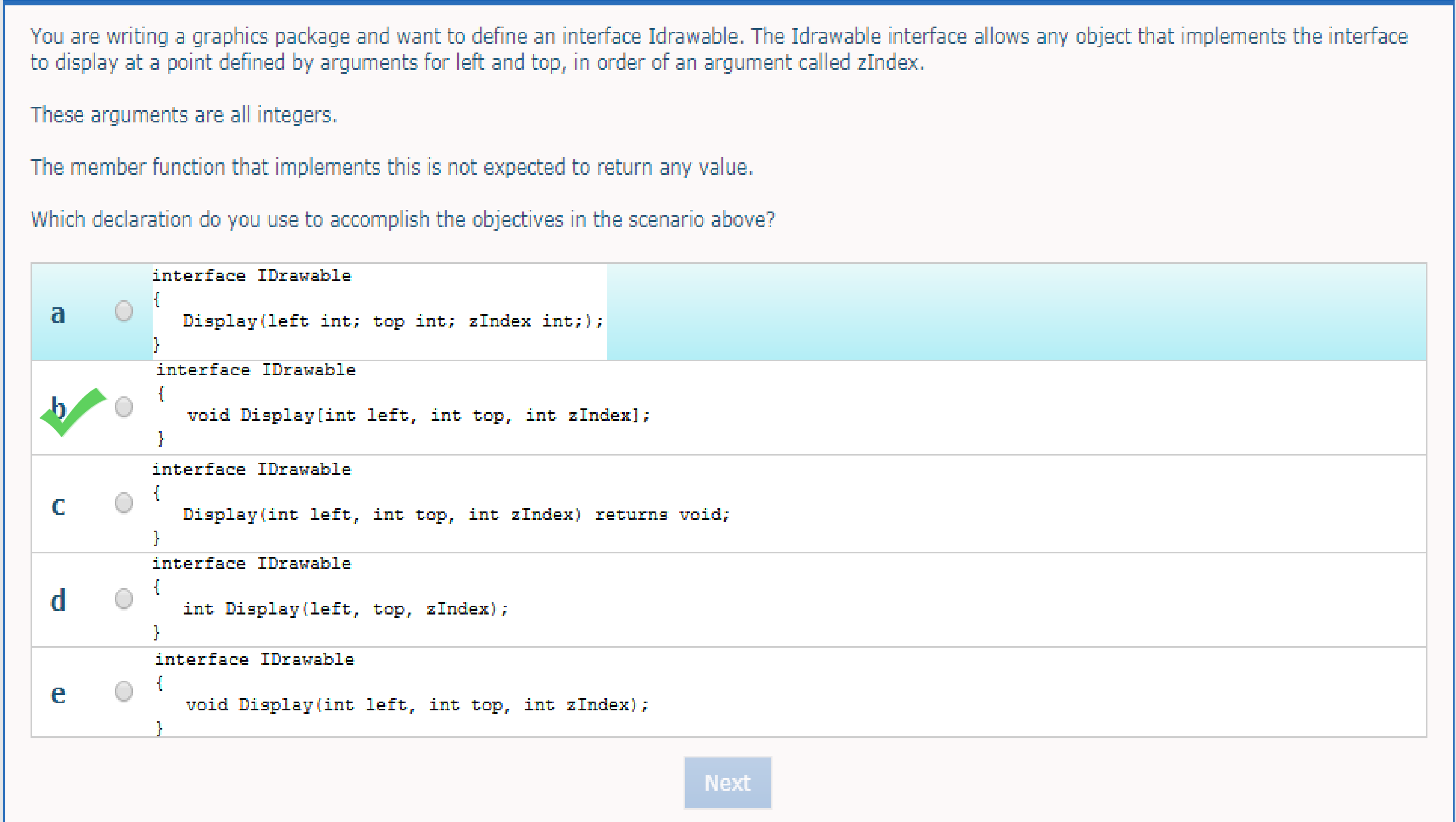
a ○ `int[] evenNumbers = Array.ForEach<int>(numbers, i => Int32.Parse(i, true));`

b ○ `int[] evenNumbers = Array.FindAll<int>(numbers, i => Int32.Parse(i, true));`

[✓ c] ○ `int[] evenNumbers = Array.ForEach<int>(numbers, i => i % 2 != 0);`

d ○ `int[] evenNumbers = Array.FindAll<int>(numbers, i => i % 2 == 0);`

e ○ `int[] evenNumbers = Array.FindAll<int>(i => i++ % 2 == 0);`



You are writing a graphics package and want to define an interface Idrawable. The Idrawable interface allows any object that implements the interface to display at a point defined by arguments for left and top, in order of an argument called zIndex.

These arguments are all integers.

The member function that implements this is not expected to return any value.

Which declaration do you use to accomplish the objectives in the scenario above?

a

```
interface IDrawable
{
   Display(left int; top int; zIndex int;);
}
```

✓ b

```
interface IDrawable
{
   void Display[int left, int top, int zIndex];
}
```

c

```
interface IDrawable
{
   Display(int left, int top, int zIndex) returns void;
}
```

d

```
interface IDrawable
{
   int Display(left, top, zIndex);
}
```

e

```
interface IDrawable
{
   void Display(int left, int top, int zIndex);
}
```

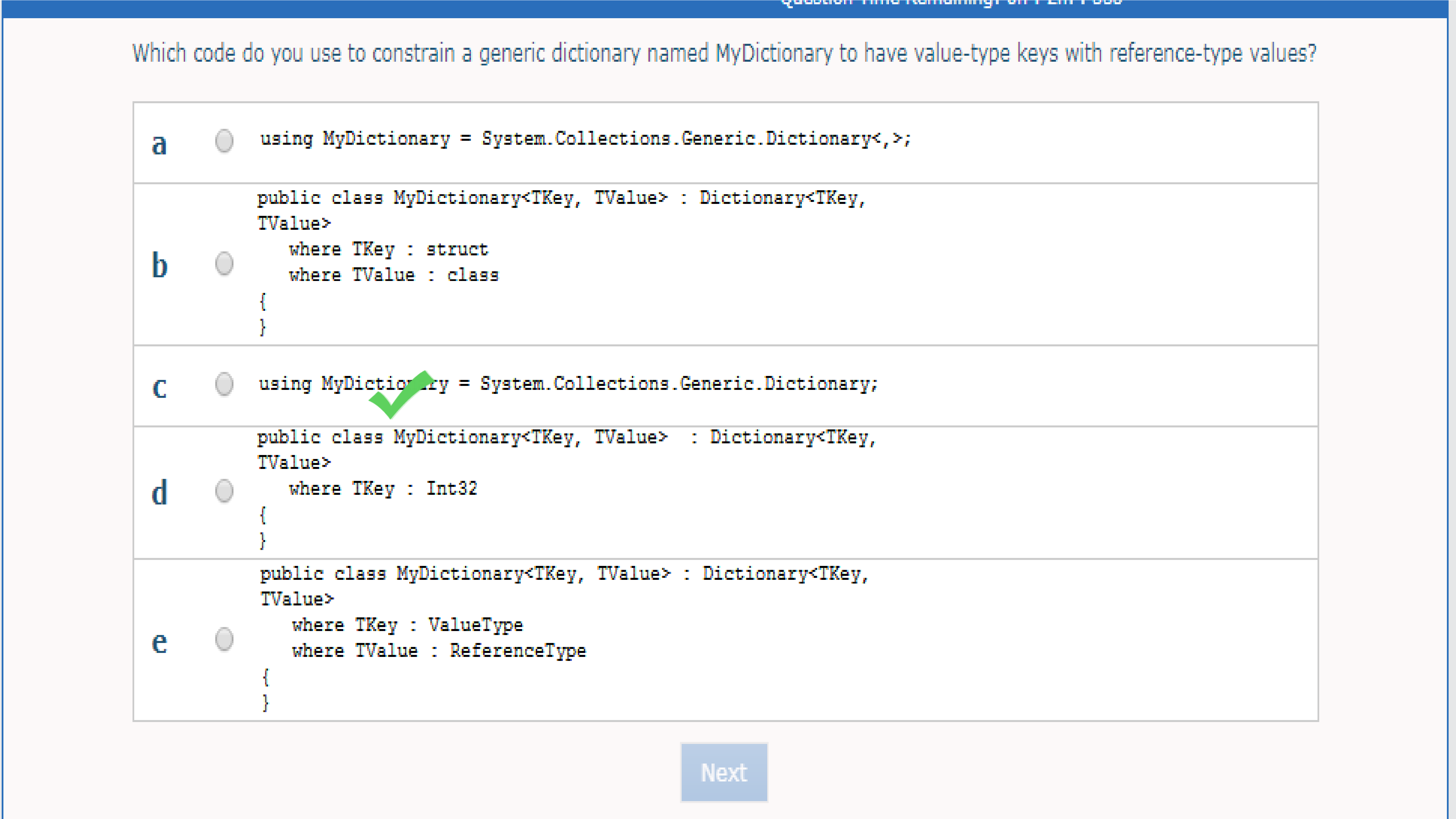
Next

```
public class Person {
 public readonly string Name;
 public readonly int Age;
 public Person(int age) {
   Age = age;
 }
 public Person NewPerson(string name, int age) {
   Person newPerson = new Person(age);
   newPerson.Name = name;
   return newPerson;
 }
}
```

Based on the sample code above, why does the code fail to compile?

✓ a	○	A readonly field can only be assigned in the constructor or at declaration.
b	○	string fields must not be readonly.
c	○	int fields must not be readonly.
d	○	The NewPerson class is not marked static.
e	○	Both int and string fields must not be readonly.

Next



Which code do you use to constrain a generic dictionary named MyDictionary to have value-type keys with reference-type values?

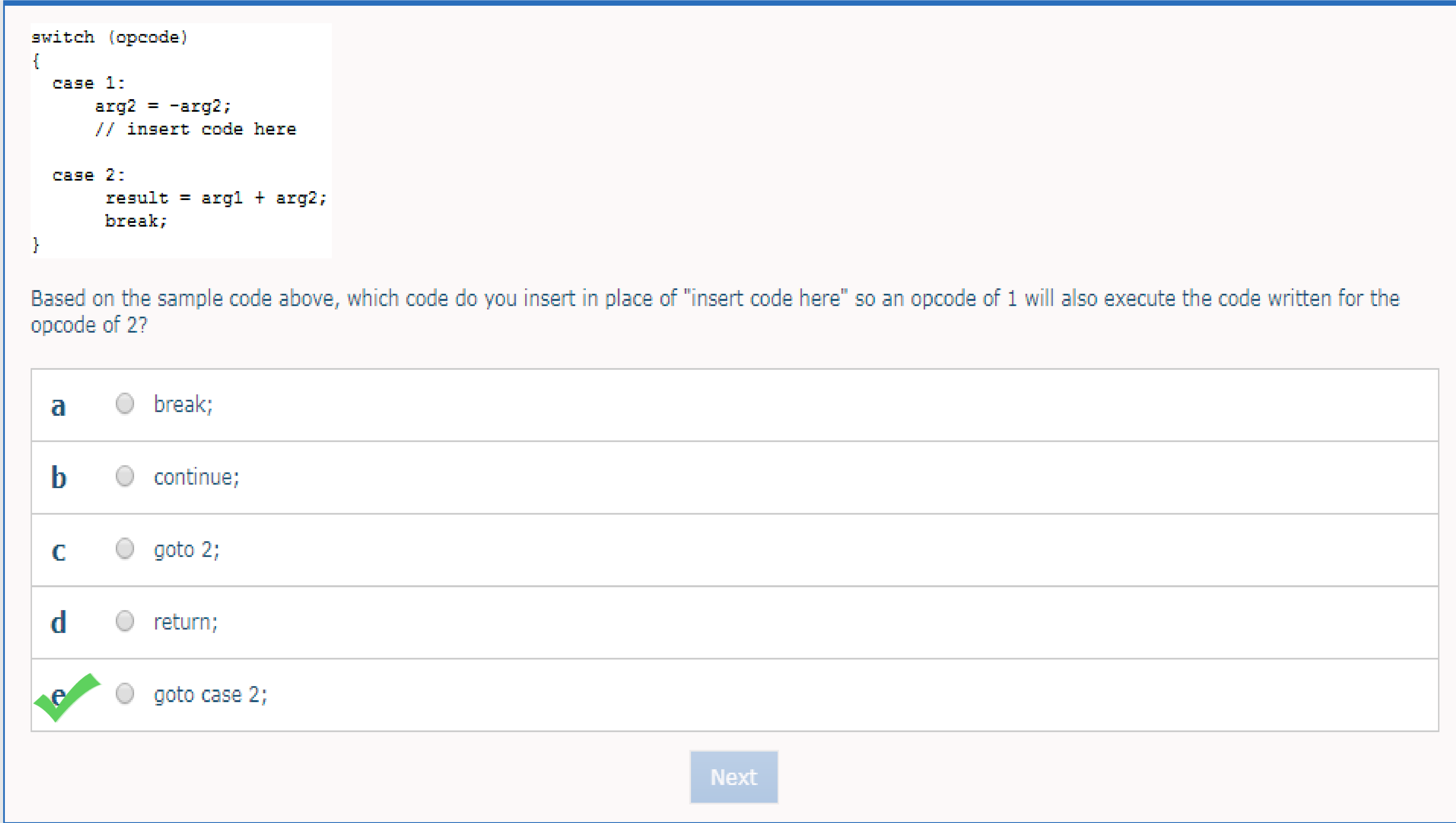
a
```
using MyDictionary = System.Collections.Generic.Dictionary<,>;
```

b
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : struct
   where TValue : class
{
}
```

[✓ c]
```
using MyDictionary = System.Collections.Generic.Dictionary;
```

d
```
public class MyDictionary<TKey, TValue>  : Dictionary<TKey,
TValue>
   where TKey : Int32
{
}
```

e
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : ValueType
   where TValue : ReferenceType
{
}
```



```
switch (opcode)
{
  case 1:
      arg2 = -arg2;
      // insert code here

  case 2:
       result = arg1 + arg2;
       break;
}
```

Based on the sample code above, which code do you insert in place of "insert code here" so an opcode of 1 will also execute the code written for the opcode of 2?

a	○ break;
b	○ continue;
c	○ goto 2;
d	○ return;
[✓ e]	○ goto case 2;

Next

```
int index = 0;
index += 1;

Label:
index += 1;
Console.WriteLine(index);
if (index < 3)
{
   goto Label;
}
```

What is written to the console when you execute the sample code above?

- a
  ```
  0
  1
  2
  ```
- b
  ```
  0
  2
  ```
- c
  ```
  1
  2
  3
  ```
- [✓ d]
  ```
  2
  3
  ```
- e
  ```
  3
  4
  ```

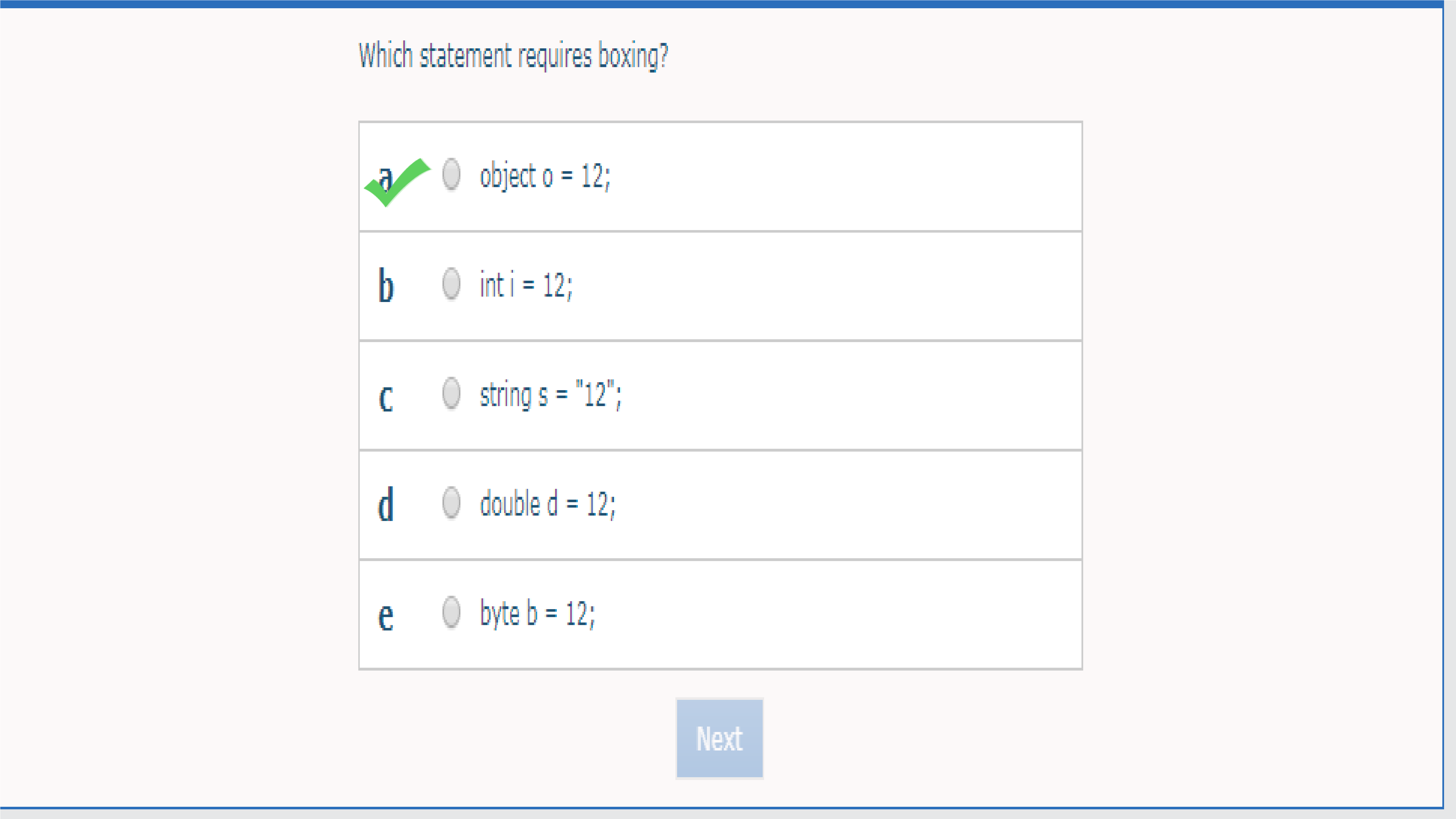
Next

```
static void Main()
    {
        var stack = new Stack<int>();
        var queue = new Queue<int>();
        foreach (var i in new int[] { 1, 2})
        {
            stack.Push(i);
            queue.Enqueue(i);
        }
        foreach (var i in stack)
        {
            Console.WriteLine(i);
        }
        foreach (var i in queue)
        {
            Console.WriteLine(i);
        }

    }
```

What is written to the console after the sample code above executes?

a	○	1 1 2 2
b	○	1 2 1 2
✓ c	○	2 1 1 2
d	○	2 1 2 1
e	○	2 2 1 1

Next



Which statement requires boxing?

[✓ a] `object o = 12;`

b `int i = 12;`

c `string s = "12";`

d `double d = 12;`

e `byte b = 12;`

You have a class "Person" that has a method named "GetIdentifier". You call this method in the class definition, and you want derived classes such as "Employee" or "BoardMember" to be able to call this method, even if these classes are derived in code added by your customer base. However, you do not want to allow this method to be called anywhere else.

In order to achieve the objectives in the scenario above, you declare the GetIdentifier method as:

a ○ protected in derived classes.

[✓ b] ○ private in Person.

c ○ internal in Person.

d ○ protected in Person.

e ○ private in derived classes.

Next

```
static object StartList(object[] array)
  {
      foreach (var o in array)
      {
          if (o is IEnumerable)
          {
              return ((IEnumerable)o).GetEnumerator();
          }
      }
      return Enumerable.Range(0, 4).GetEnumerator();
  }
```

Which interface does the object that is returned by the method in the sample code above implement?

a	IEnumerable
[✓ b]	IEnumerator
c	IArray
d	IList
e	ICollection

Next

```
static void Main()
{
   Compound x = new Compound();
   x.Mix(ref x.NumberOfTimesMixed);
   Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
   public int NumberOfTimesMixed { get; set;}

   public void Mix(ref int counter, params Compound[] components)
   {
        counter++;
        // further processing
   }
   }
```

How do you fix the sample code above?

✓ a ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
int i = x.NumberOfTimesMixed;
x.Mix(ref i);
x.NumberOfTimesMixed = i;
```

b ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
x.Mix(x.NumberOfTimesMixed);
```

c ○ Insert

```
public Compound() {}
```

above

```
public int NumberOfTimesMixed { get; set;}
```

d ○ Change

```
x.Mix(ref x.NumberOfTimesMixed);
```

to

```
x.Mix(ref x.NumberOfTimesMixed, x);
```

e ○ Remove

```
public int NumberOfTimesMixed { get; set;}
```

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

a

```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

b

```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

✓ c

```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

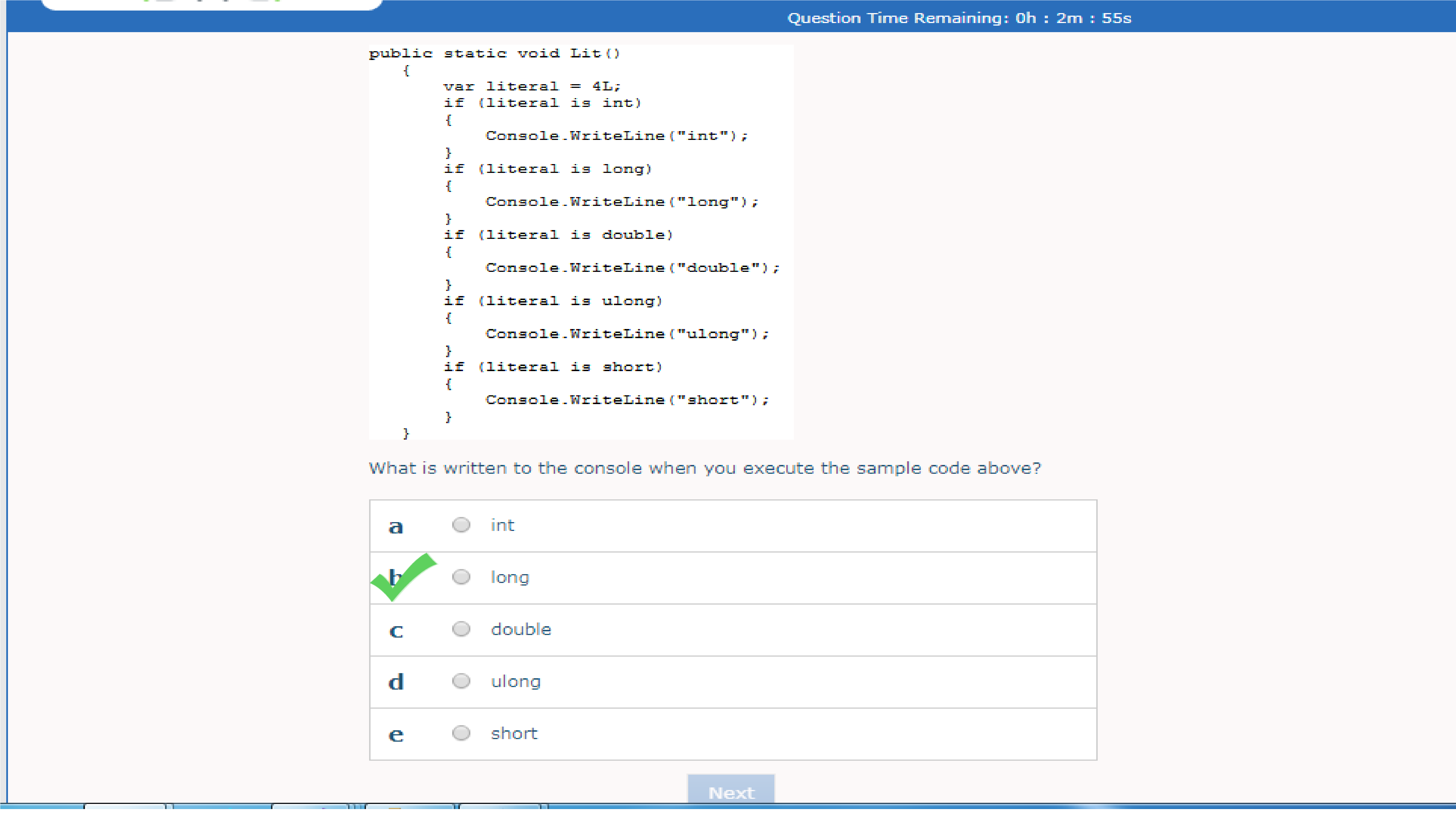
d

```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

e

```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```

Next



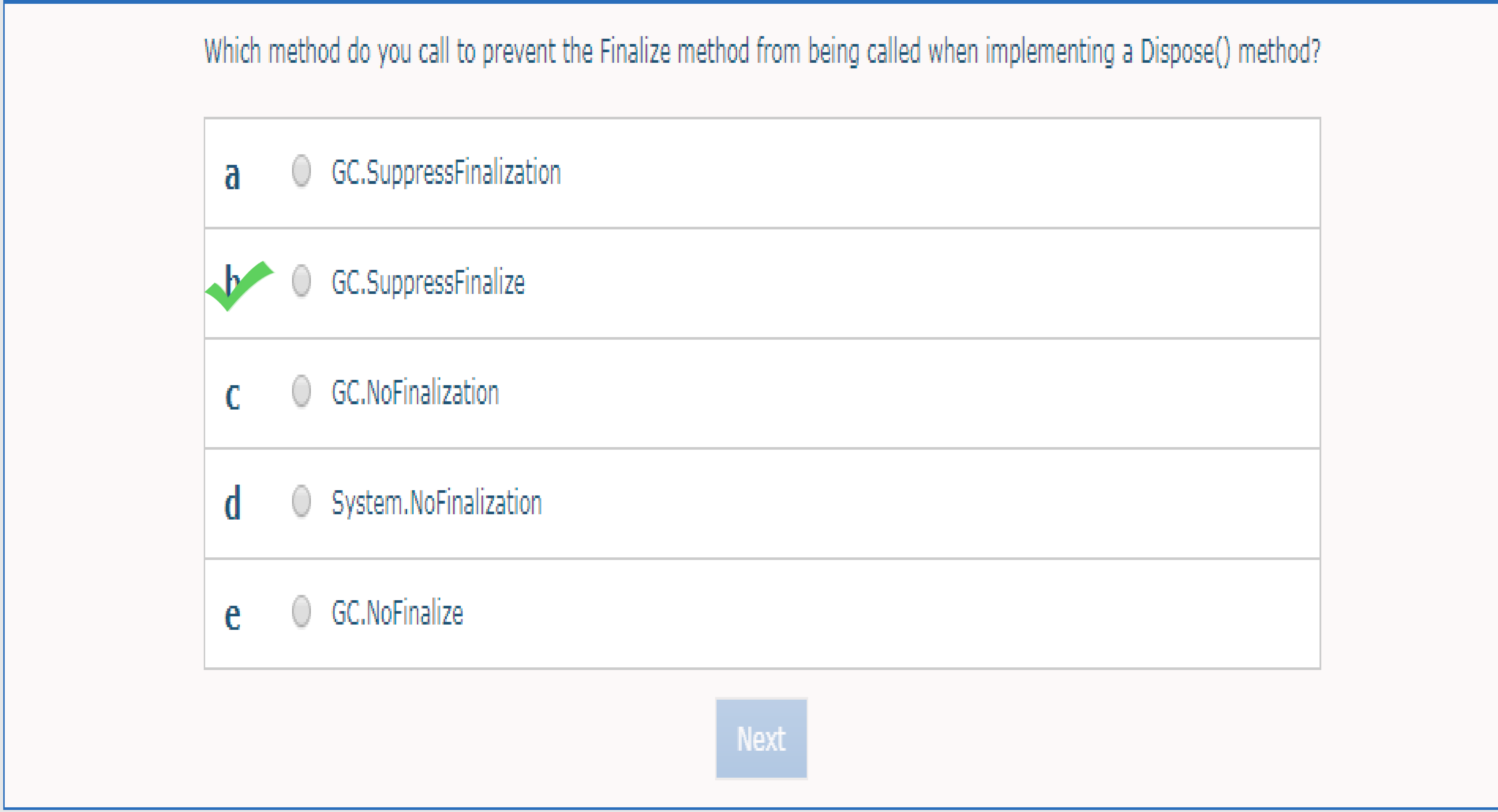
Question Time Remaining: 0h : 2m : 55s

```
public static void Lit()
    {
         var literal = 4L;
         if (literal is int)
         {
              Console.WriteLine("int");
         }
         if (literal is long)
         {
              Console.WriteLine("long");
         }
         if (literal is double)
         {
              Console.WriteLine("double");
         }
         if (literal is ulong)
         {
              Console.WriteLine("ulong");
         }
         if (literal is short)
         {
              Console.WriteLine("short");
         }
    }
```

What is written to the console when you execute the sample code above?

a	○	int
[✓ b]	○	long
c	○	double
d	○	ulong
e	○	short

Next



Which method do you call to prevent the Finalize method from being called when implementing a Dispose() method?

a	GC.SuppressFinalization
[✓ b]	GC.SuppressFinalize
c	GC.NoFinalization
d	System.NoFinalization
e	GC.NoFinalize

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

[✓ a]	Task context = a => { outputText.Text = a};
b	TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;
c	TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();
d	TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;
e	CancellationToken context = new CancellationToken();

```
private void MyMethod(string param1, string param2);
```

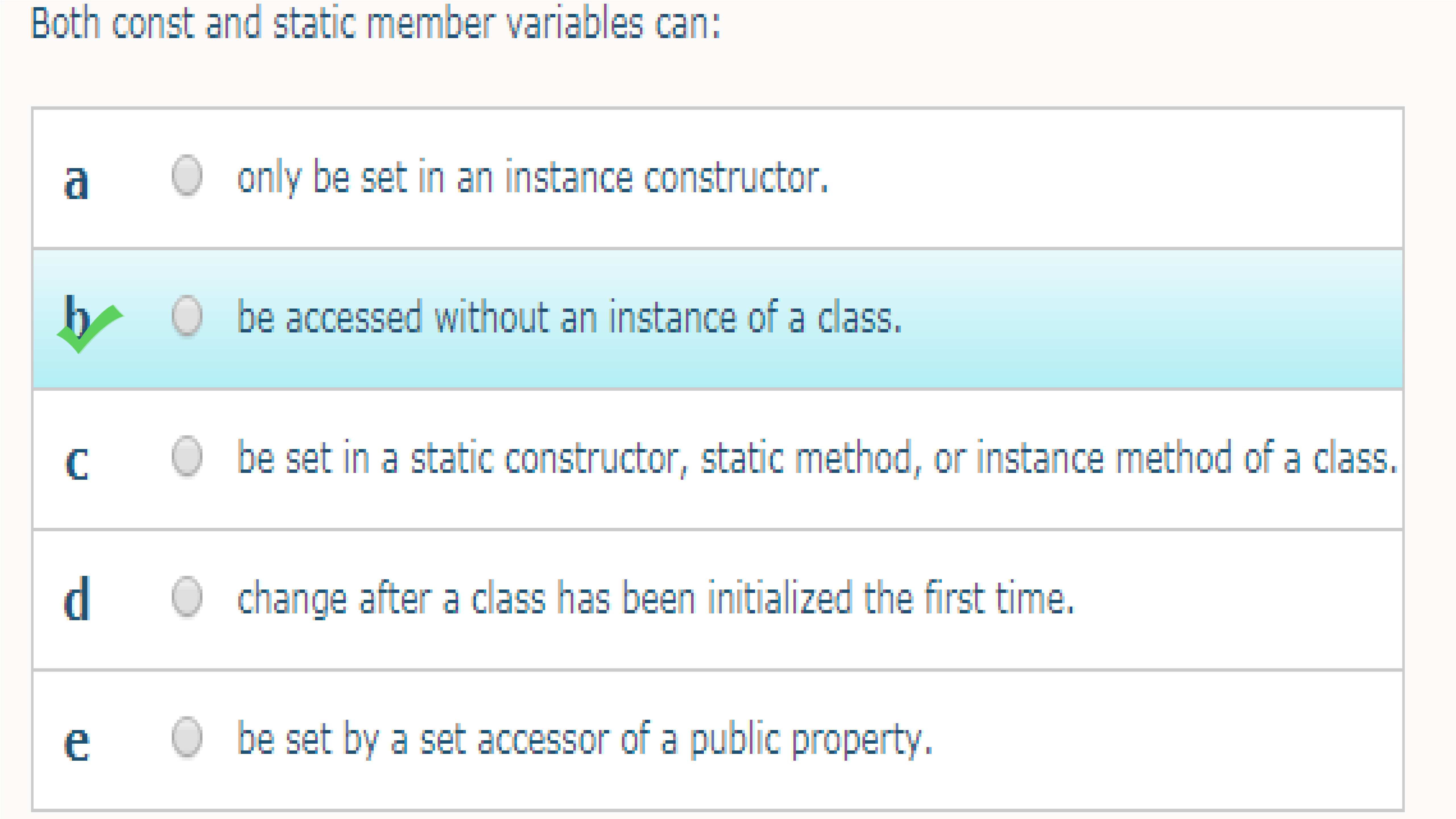
Based on the sample code above, how do you make param2 optional and set a default value of null?

a	○ Only add DefaultValue("param2", null) as an attribute to the method.
b	○ Change string param2 in the method signature to string param2 = null.
[✓ c]	○ Add both DefaultValue("param2", null) and OptionalValue("param2") as attributes.
d	○ Only add DefaultValue("string param2", null) as an attribute to the method.
e	○ Change string param2 in the method signature to [string param2 = null].

```
// insert declaration
{
   WebRequest myRequest = WebRequest.Create(url);
   WebResponse r = await myRequest.GetResponseAsync();
   StreamReader sr = new StreamReader( r.GetResponseStream() );
   string text = sr.ReadToEnd();
   // do some processing of text, details omitted
}
```

Based on the sample code above, which declaration do you use in place of // insert declaration?

a	task void Process()
b	async void Process()
c	void await Process()
d	void partial Process()
✓ e	void Process()



Both const and static member variables can:

a	only be set in an instance constructor.
[✓ b]	be accessed without an instance of a class.
c	be set in a static constructor, static method, or instance method of a class.
d	change after a class has been initialized the first time.
e	be set by a set accessor of a public property.

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

a

```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

b

```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

c

```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

d

```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

[✓ **e**]

```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of t
property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

a

```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```

[✓ b]

```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

c

d

```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

e

```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

- **a** ○ CancellationToken context = new CancellationToken();
- **b** ○ TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;
- [✓ **c**] ○ Task context = a => { outputText.Text = a};
- **d** ○ TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();
- **e** ○ TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;

```
static void Main()
{
    Compound x = new Compound();
    x.Mix(ref x.NumberOfTimesMixed);
    Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
    public int NumberOfTimesMixed { get; set;}

    public void Mix(ref int counter, params Compound[] components)
    {
         counter++;
         // further processing
    }
    }
```

How do you fix the sample code above?

a ○ Change
```
x.Mix(ref x.NumberOfTimesMixed);
```
to
```
x.Mix(x.NumberOfTimesMixed);
```

b ○ Insert
```
public Compound() {}
```
above
```
public int NumberOfTimesMixed { get; set;}
```

[✓ c] ○ Change
```
x.Mix(ref x.NumberOfTimesMixed);
```
to
```
int i = x.NumberOfTimesMixed;
x.Mix(ref i);
x.NumberOfTimesMixed = i;
```

d ○ Change
```
x.Mix(ref x.NumberOfTimesMixed);
```
to
```
x.Mix(ref x.NumberOfTimesMixed, x);
```

e ○ Remove
```
public int NumberOfTimesMixed { get; set;}
```

```
static object StartList(object[] array)
   {
       foreach (var o in array)
       {
           if (o is IEnumerable)
           {
               return ((IEnumerable)o).GetEnumerator();
           }
       }
       return Enumerable.Range(0, 4).GetEnumerator();
   }
```

Which interface does the object that is returned by the method in the sample code above implement?

[✓ a] IEnumerator

b IEnumerable

c IList

d ICollection

e IArray

Next

You are writing a method that has a "catch" block for the System.StackOverflowException. The block stores the original exception in the variable "exc". Within this block the method frees some resources, writes a message to the console, and then rethrows the original exception without losing any information.

Based on the scenario above, which statement do you use to rethrow the exception?

✓ a	○	recatch new StackOverflowException();
b	○	finally exc;
c	○	throw;
d	○	throw new StackOverflowException();
e	○	rethrow exc;

```
1: var lst = new List<object>();
2: DateTime start = DateTime.Now;
3: for (int counter = 0; counter < 10000000; counter++)
4: {
5:     lst.Add(counter);
6: }
7: Console.WriteLine((DateTime.Now - start).TotalMilliseconds);
```

Which of the following changes do you implement to speed up the sample code above by a factor of 10?

a ○ Change line 1 to:
var lst = new IEnumerable<int>();

b ✓ ○ Change line 1 to:
var lst = new List<int>();

c ○ Between lines 4 and 5, insert:
lst.Sort();

d ○ Change line 5 to:
lst.Insert(counter, counter);

e ○ Change line 5 to:
lst.Add((object)counter);

```
var list = new IList<int>();
```

Which do you change in the sample code above for it to compile?

a	○ IList<int>() to IList();
b	○ new IList<int>() to IList<int>.CreateNew();
[✓ c]	○ var to IList<int>
d	○ the variable name list to l
e	○ IList<int>() to List<int>();

```
object o = null;
try {
    int? i = (int?)o;
    int i2 = i ?? 0;
    Console.WriteLine(i2);
}
catch(Exception ex){
    Console.WriteLine(ex.Message);
}
```

Based on the sample code above, what is written to the console?

a	○	A blank line
b	○	Object reference not set to an instance of an object.
c	○	ArgumentException
d	○	null
[✓ e]	○	0

You are creating a list of strings named "types" that is initialized with the names of the primitive types "int", "long", and "short", in that order. You have already created a "using" directive for System.Collections.Generic.

Which line of code do you use to accomplish the objective in the scenario above?

a ○ `List<string> types = { "int", "long", "short" };`

[✓ b] ○ `List<string> types = new List<string> { "int", "long", "short"};`

c ○ `List<string> types = new [ "int", "long", "short" ];`

d ○ `List<string> types = [ "int", "long", "short" ];`

e ○ `List<string> types = new List<string>( "int", "long", "short");`

You use P/Invoke to access:

a ✓	○	functions defined in Win32 DLLs.
b	○	strings representing Pascal code.
c	○	events in other assemblies.
d	○	delegates in other assemblies.
e	○	methods compiled for nonstandard CPUs.

```
Snippet A:
string s1 = "1";
string s2 = s1;

Snippet B:
int i = 1;
string s2 = i.ToString();
```

Based on the sample code above, why is snippet A faster than snippet B?

a	Boxing only impacts performance when /unsafe is not specified.
b	Boxed reference types are faster than using value types.
c	Boxing is faster than a reference assignment.
d	Boxed value types are faster than using reference types.
[✓ e]	Boxing is slower than a reference assignment.

Which keyword do you use to ensure multiple threads can access the current value of a variable at all times?

a	checked
b	unsafe
c	extern
d	static
[✓ e]	volatile

```
public class Employee
{
   public virtual void Hire() { /* implement */}
   public virtual void Terminate() { /* implement */}
}

public class Manager : Employee
{
   public override void Hire() { base.Hire(); /* implement more */}
   // implement Terminate
}
```

Based on the sample code above, which declaration do you use for Terminate in the class Manager so no class deriving from Manager will override it?

a ○ public final override void Terminate()

[✓ b] ○ public static void Terminate()

c ○ public volatile void Terminate()

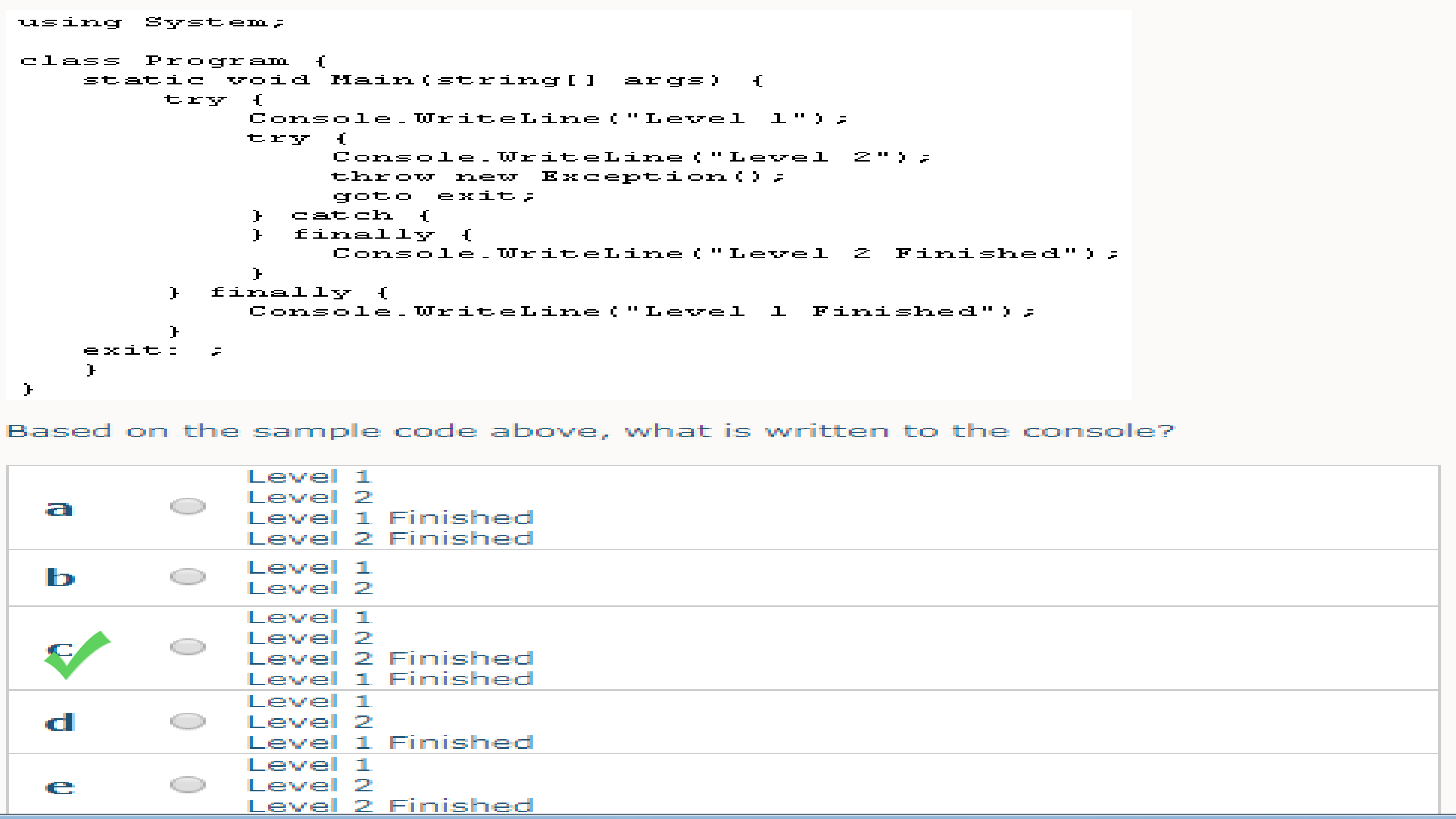
d ○ public sealed override void Terminate()

e ○ public virtual void Terminate()

```
// insert declaration
{
  WebRequest myRequest = WebRequest.Create(url);
  WebResponse r = await myRequest.GetResponseAsync();
  StreamReader sr = new StreamReader( r.GetResponseStream() );
  string text = sr.ReadToEnd();
  // do some processing of text, details omitted
}
```

Based on the sample code above, which declaration do you use in place of // insert declaration?

a	○	void await Process()
b	○	void partial Process()
✓ c	○	void Process()
d	○	task void Process()
e	○	async void Process()



```
using System;

class Program {
    static void Main(string[] args) {
        try {
            Console.WriteLine("Level 1");
            try {
                Console.WriteLine("Level 2");
                throw new Exception();
                goto exit;
            } catch {
            } finally {
                Console.WriteLine("Level 2 Finished");
            }
        } finally {
            Console.WriteLine("Level 1 Finished");
        }
    exit: ;
    }
}
```

Based on the sample code above, what is written to the console?

a	○	Level 1 Level 2 Level 1 Finished Level 2 Finished
b	○	Level 1 Level 2
[✓ c]	○	Level 1 Level 2 Level 2 Finished Level 1 Finished
d	○	Level 1 Level 2 Level 1 Finished
e	○	Level 1 Level 2 Level 2 Finished

You are creating an Ingredient class, and you want to provide developers a way to combine two Ingredients, forming a new instance of Ingredient. You want to be able to write:

```
public Ingredient Mix(Ingredient a, Ingredient b)
{
  return a + b;
}
```

Based on the scenario above, how do you satisfy the requirements?

a	○	Override the Equals and GetHashCode methods on the Ingredient class.
b	○	Overload the + operator in the Ingredient class.
✓ c	○	Implement the IMergeable interface.
d	○	Change the Ingredient class to a struct.
e	○	Create a new class called NewIngredient with a constructor that accepts two Ingredients.

You are creating an Ingredient class, and you want to provide developers a way to combine two Ingredients, forming a new instance of Ingredient. You want to be able to write:

```
public Ingredient Mix(Ingredient a, Ingredient b)
{
  return a + b;
}
```

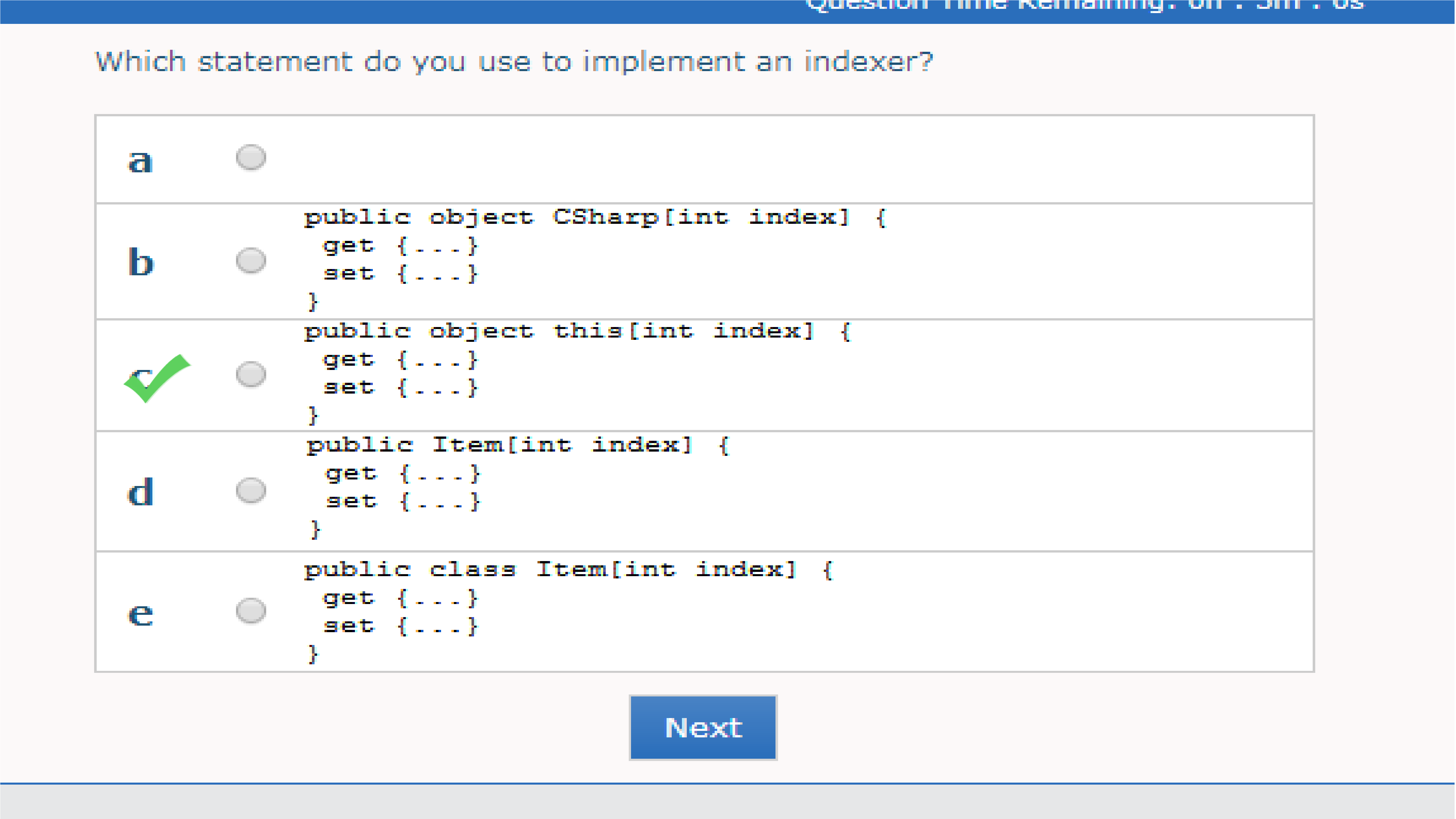
Based on the scenario above, how do you satisfy the requirements?

a	○	Override the Equals and GetHashCode methods on the Ingredient class.
b ✓	○	Overload the + operator in the Ingredient class.
c	○	Implement the IMergeable interface.
d	○	Change the Ingredient class to a struct.
e	○	Create a new class called NewIngredient with a constructor that accepts two Ingredients.

You are writing a compiler and need to keep track of all the identifiers created by the user. Whenever an identifier comes up in the code you are compiling, you need to quickly access the declaration of that identifier via its string value, even if a large number of identifiers are in use.

Based on the scenario above, which data structure do you use to meet the requirements?

- a ○ List<string>
- [✓ b] ○ Stack
- c ○ Queue
- d ○ ArrayList
- e ○ Hashtable



Which statement do you use to implement an indexer?

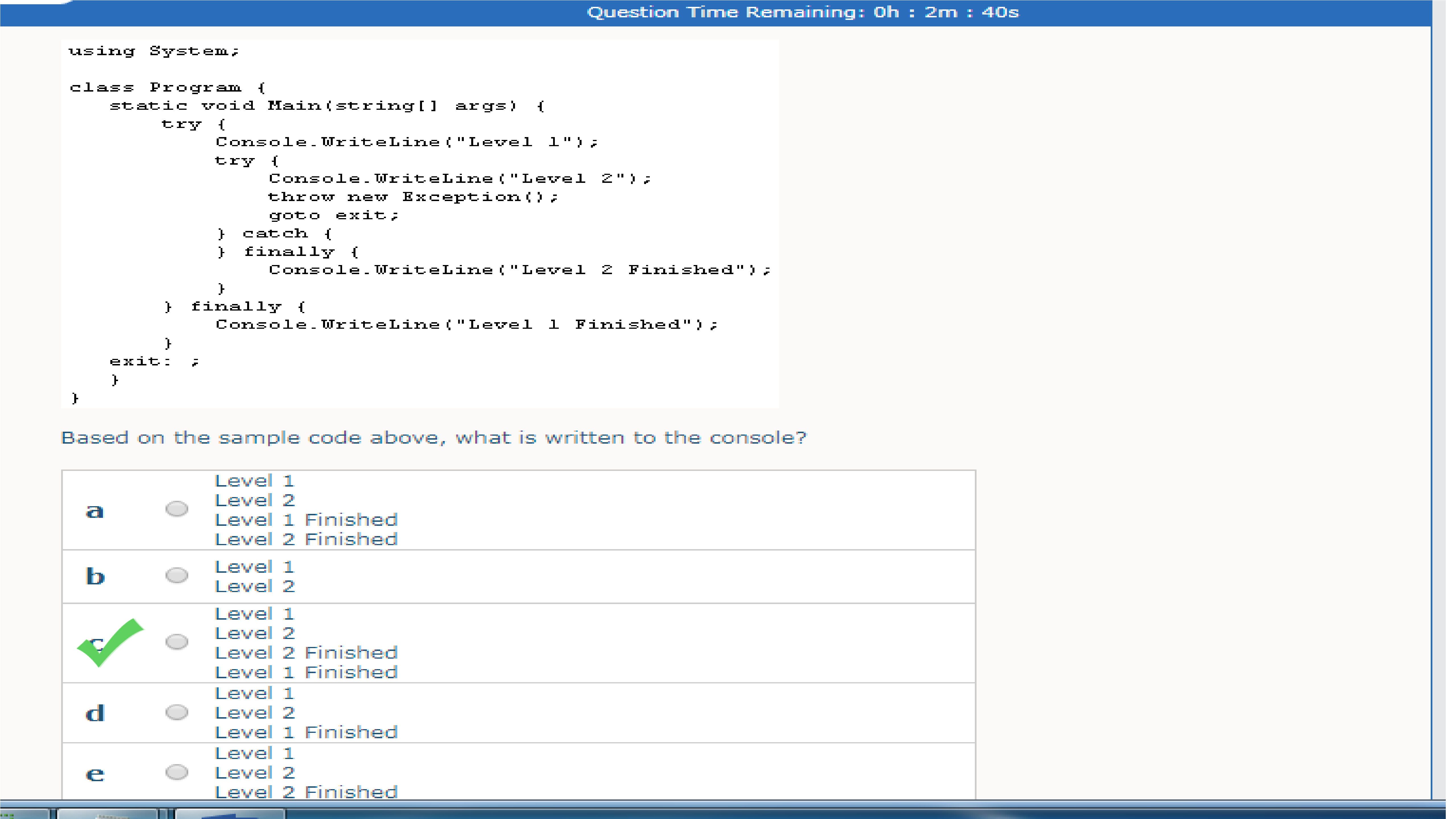
a

b
```
public object CSharp[int index] {
 get {...}
 set {...}
}
```

[✓ c]
```
public object this[int index] {
 get {...}
 set {...}
}
```

d
```
public Item[int index] {
 get {...}
 set {...}
}
```

e
```
public class Item[int index] {
 get {...}
 set {...}
}
```

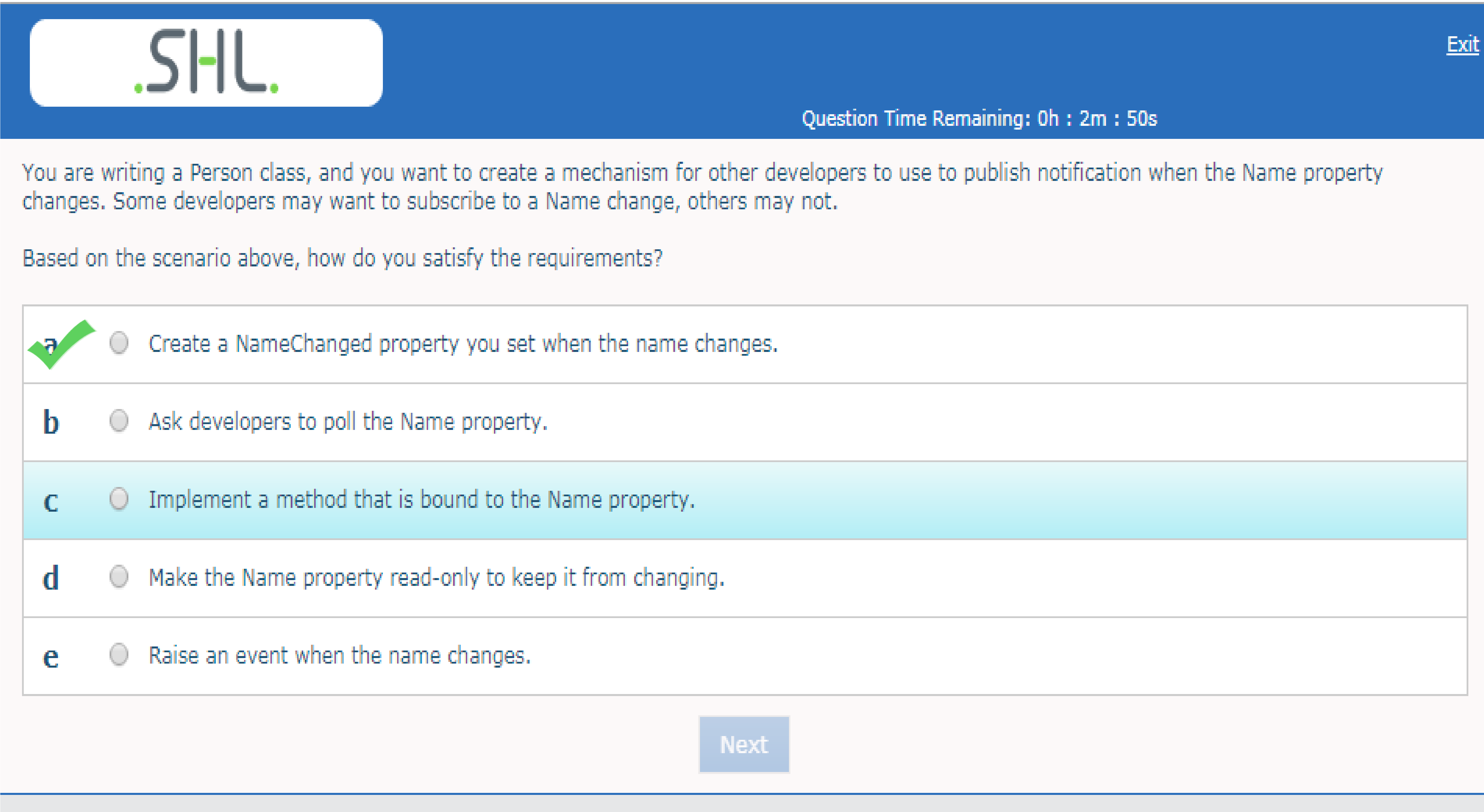


Question Time Remaining: 0h : 2m : 40s

```
using System;

class Program {
    static void Main(string[] args) {
        try {
            Console.WriteLine("Level 1");
            try {
                Console.WriteLine("Level 2");
                throw new Exception();
                goto exit;
            } catch {
            } finally {
                Console.WriteLine("Level 2 Finished");
            }
        } finally {
            Console.WriteLine("Level 1 Finished");
        }
    exit: ;
    }
}
```

Based on the sample code above, what is written to the console?

a	○	Level 1 Level 2 Level 1 Finished Level 2 Finished
b	○	Level 1 Level 2
[✓ c]	○	Level 1 Level 2 Level 2 Finished Level 1 Finished
d	○	Level 1 Level 2 Level 1 Finished
e	○	Level 1 Level 2 Level 2 Finished



nline.eu/default.action=testquestion&testaction=-1&timetaken=101890&audiopage=false&dbgs=19401163

Exit

Question Time Remaining: 0h : 2m : 50s

You are writing a Person class, and you want to create a mechanism for other developers to use to publish notification when the Name property changes. Some developers may want to subscribe to a Name change, others may not.

Based on the scenario above, how do you satisfy the requirements?

✓ a	○	Create a NameChanged property you set when the name changes.
b	○	Ask developers to poll the Name property.
c	○	Implement a method that is bound to the Name property.
d	○	Make the Name property read-only to keep it from changing.
e	○	Raise an event when the name changes.

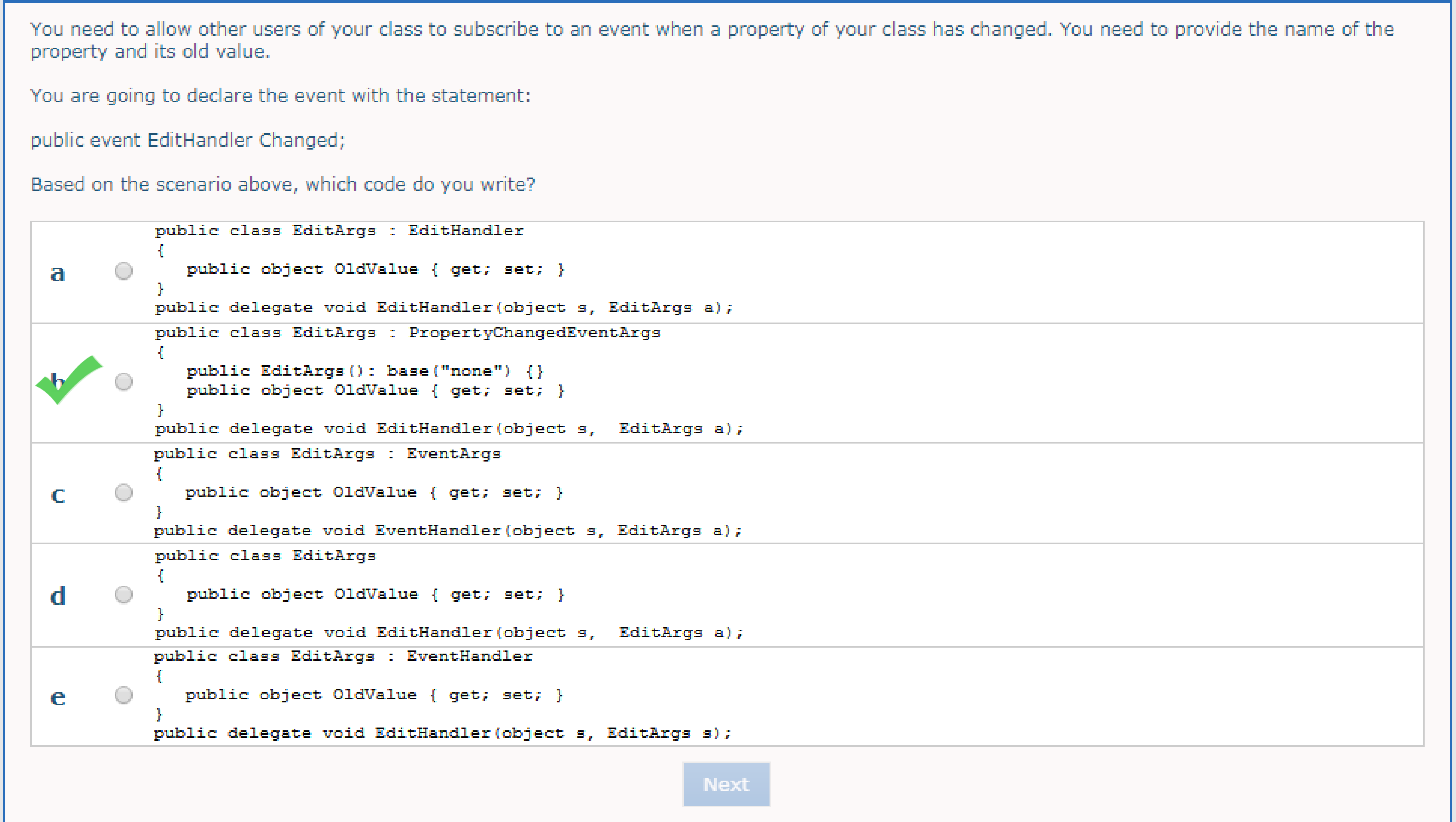
Next

```
private void ProcessAll()
{
// insert code here
   foreach (var i in listBox1.Items)
   {
       string url = i.ToString();
       WebRequest r = WebRequest.Create(url);
       Task<WebResponse> t = r.GetResponseAsync();
       t.ContinueWith(x => {
           outputText.Text =
               (new StreamReader(x.Result.GetResponseStream())).
               ReadToEnd(); },
               context);
   }
}
```

The method in the sample code above is called from the UI thread. Which code do you insert in "// insert code here" to ensure each web request is written to the "outputText" text box as it completes?

a	○ Task context = a => { outputText.Text = a};
b	○ TaskScheduler context = TaskScheduler.FromCurrentSynchronizationContext();
c	○ TaskContinuationOptions context = TaskContinuationOptions.ExecuteSynchronously;
[✓ d]	○ CancellationToken context = new CancellationToken();
e	○ TaskContinuationOptions context = TaskContinuationOptions.PreferFairness;

Next



You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

a

```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

[✓ b]

```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

c

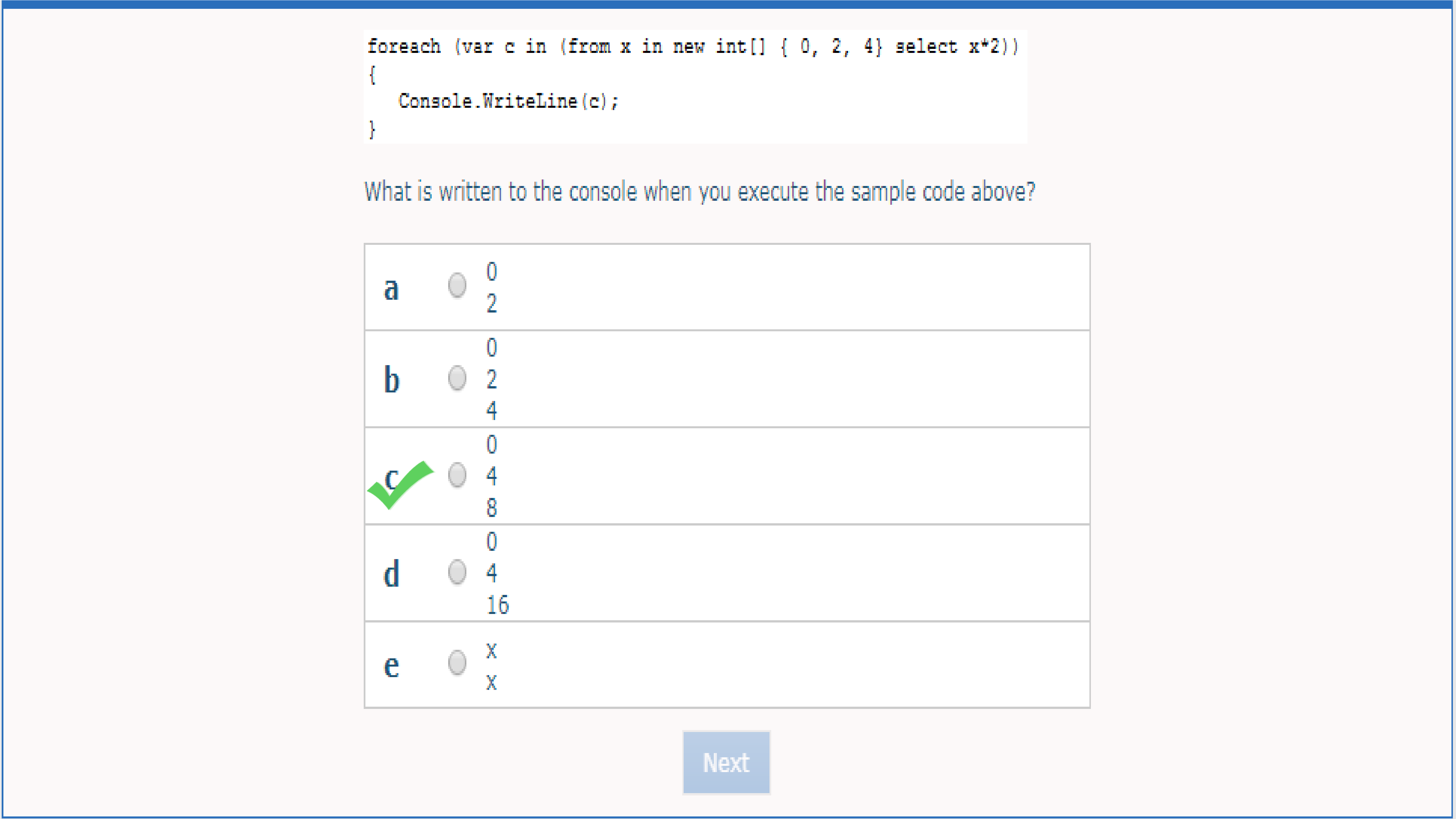
```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```

d

```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

e

```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```



```
foreach (var c in (from x in new int[] { 0, 2, 4} select x*2))
{
   Console.WriteLine(c);
}
```

What is written to the console when you execute the sample code above?

a	0 2
b	0 2 4
[✓ c]	0 4 8
d	0 4 16
e	x x

Which method do you implement to create a Reverse() extension method to the string class?

[✓ a ● public static string Reverse(this string s);]

b ○ public static this Reverse(this string s);

c ○ public static string Reverse();

d ○ public void Reverse(this string s);

e ○ public string Reverse(string s);

```
static void ProcessWordcounts(Dictionary<string, int> counts)
{
    foreach (var element in counts)
    {
        // Perform operations on element......
    }
}
```

In the sample code above, what is the datatype of element?

a	○	IEnumerable<int>
b	○	string
c	○	Dictionary<string, int>
✓ d	○	KeyValuePair<string, int>
e	○	int

Which code do you use to constrain a generic dictionary named MyDictionary to have value-type keys with reference-type values?

a

```
using MyDictionary = System.Collections.Generic.Dictionary;
```

b

```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : ValueType
   where TValue : ReferenceType
{
}
```

c

```
public class MyDictionary<TKey, TValue>  : Dictionary<TKey,
TValue>
   where TKey : Int32
{
}
```

[✓ d]

```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : struct
   where TValue : class
{
}
```

e

```
using MyDictionary = System.Collections.Generic.Dictionary<,>;
using System;

class Program {
    public static void Write(out int x) {
        Console.WriteLine(x);
        x = 1;
    }

    static int Main(string[] args) {
        int y = 3;
        Write(out y);
        return 0;
    }
}
```

Based on the sample code above, what causes the compile-time error?

a	○	The variable y is given a value prior to calling the Write method.
b	○	The y variable must be explicitly converted to a string when calling the Write method of the Console class.
c	○	The Write method does not have a return keyword in its body.
✓ d	○	The Write method tries to write an unassigned variable to the console.
e	○	The Main method cannot be declared as static.

```
enum MaterialColors
{
    Blue = 1,
    Red,
    Yellow = 4,
    Purple = Blue | Red,
    Green = Yellow | Blue,
    Orange = Red | Yellow,
}

static void Main(string[] args) {
    Console.WriteLine((int)MaterialColors.Orange);
}
```

Based on the sample code above, what is written to the console?

a	○	0
b	○	4
c	○	5
[✓ d]	○	6
e	○	7

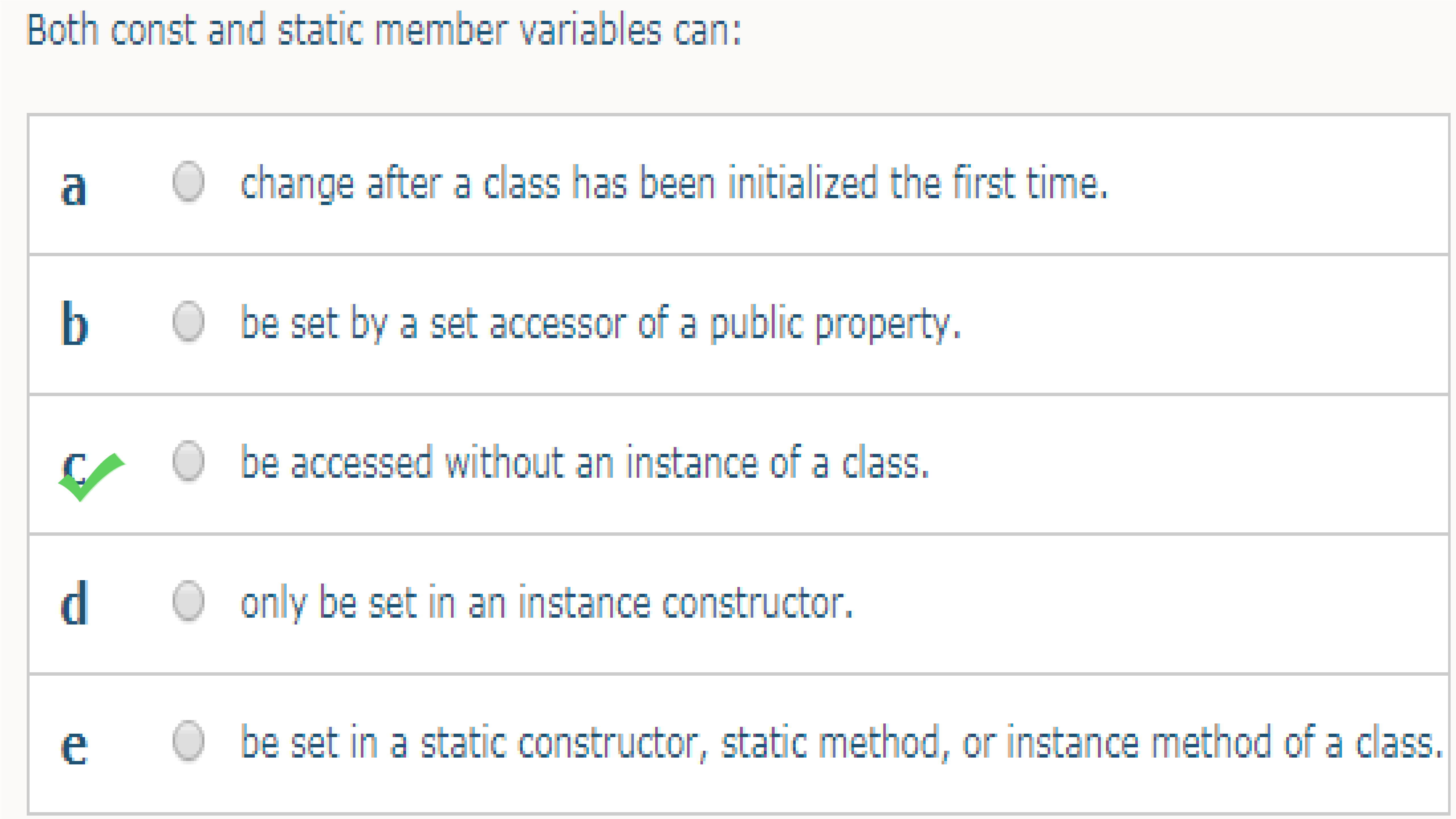
```
public static IEnumerable ParseCodes(string value, out int code)
{
    int parsedCode;
    code = 0;

    if (!Int32.TryParse(value, out parsedCode)) {
        yield return "false";
    }

    code = parsedCode;

    for (int i = 0; i <= parsedCode.ToString( ).Length; i++) {
        yield return parsedCode.ToString( ).Substring(i, 1);
    }
}

static void Main(string[] args) {
    int code;
    foreach (string n in ParseCodes("[82738]", out code)) {
        Console.WriteLine(n);
    }

    Console.WriteLine("Code: " + code);
    Console.ReadLine( );
}
```

Why is there an error when you compile the sample code above?

- a ○ Main must be declared public.
- b ○ Return cannot follow yield.
- c ○ TryParse is not a valid method.
- [✓ d] ○ You cannot use ref/out parameters with an iterator.

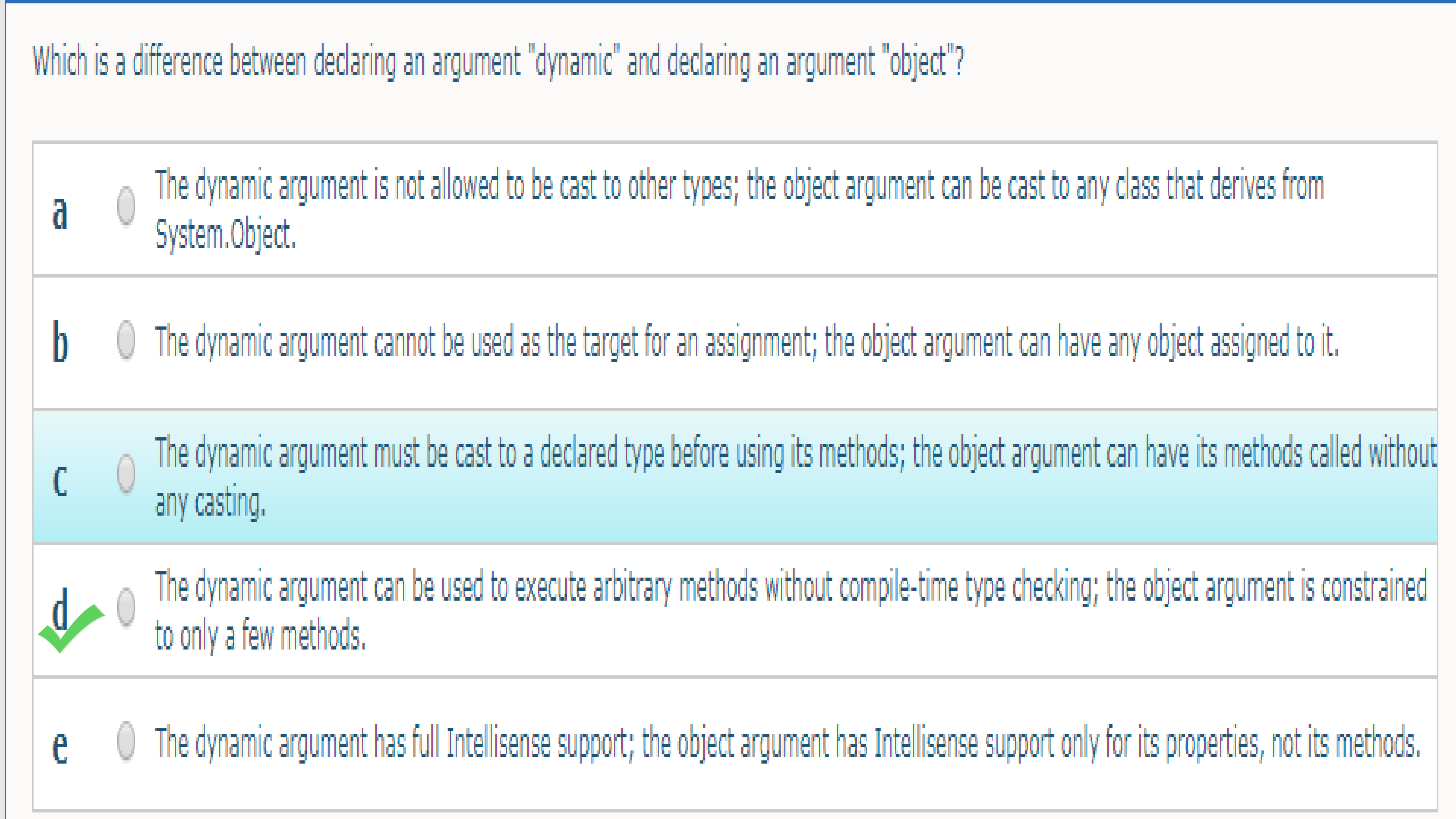
You use the Data property of an exception to:

a	○ tell the runtime the fault message of the exception.
b	○ automatically report the line of code which threw the exception.
c	○ send Microsoft special information about the application failure.
[✓ d]	○ store information that may be accessed up the call stack.
e	○ query metadata automatically in order to generate runtime.



Both const and static member variables can:

a	○	change after a class has been initialized the first time.
b	○	be set by a set accessor of a public property.
[✓ c]	○	be accessed without an instance of a class.
d	○	only be set in an instance constructor.
e	○	be set in a static constructor, static method, or instance method of a class.



Which is a difference between declaring an argument "dynamic" and declaring an argument "object"?

a ○ The dynamic argument is not allowed to be cast to other types; the object argument can be cast to any class that derives from System.Object.

b ○ The dynamic argument cannot be used as the target for an assignment; the object argument can have any object assigned to it.

c ○ The dynamic argument must be cast to a declared type before using its methods; the object argument can have its methods called without any casting.

[✓ d] ○ The dynamic argument can be used to execute arbitrary methods without compile-time type checking; the object argument is constrained to only a few methods.

e ○ The dynamic argument has full Intellisense support; the object argument has Intellisense support only for its properties, not its methods.

Which declaration do you use for a method that can be called within the assembly or any derived classes within the same assembly?

a ○ public override static void Method()

b ○ public abstract static void Method()

c ○ public internal static void Method()

d ○ private internal virtual void Method()

e ✓ ○ protected internal override void Method()

```
public class Isotope
{
    public Isotope(int number, int weight)
    {
        // initialize isotope appropriately
    }
}

public class Element
{
    public const Isotope Deuterium = new Isotope(1, 2);
}
```

Why does the sample code above fail to compile?

a	○	Element does not have a constructor.
b	○	Const members must be declared with get and set accessors.
c	○	Initializing a const requires a parameter-less constructor.
[✓ d]	○	Deuterium is not being initialized with a constant expression.
e	○	Isotope does not have a constructor.

Next

Based on the sample code above, what is written to the console?

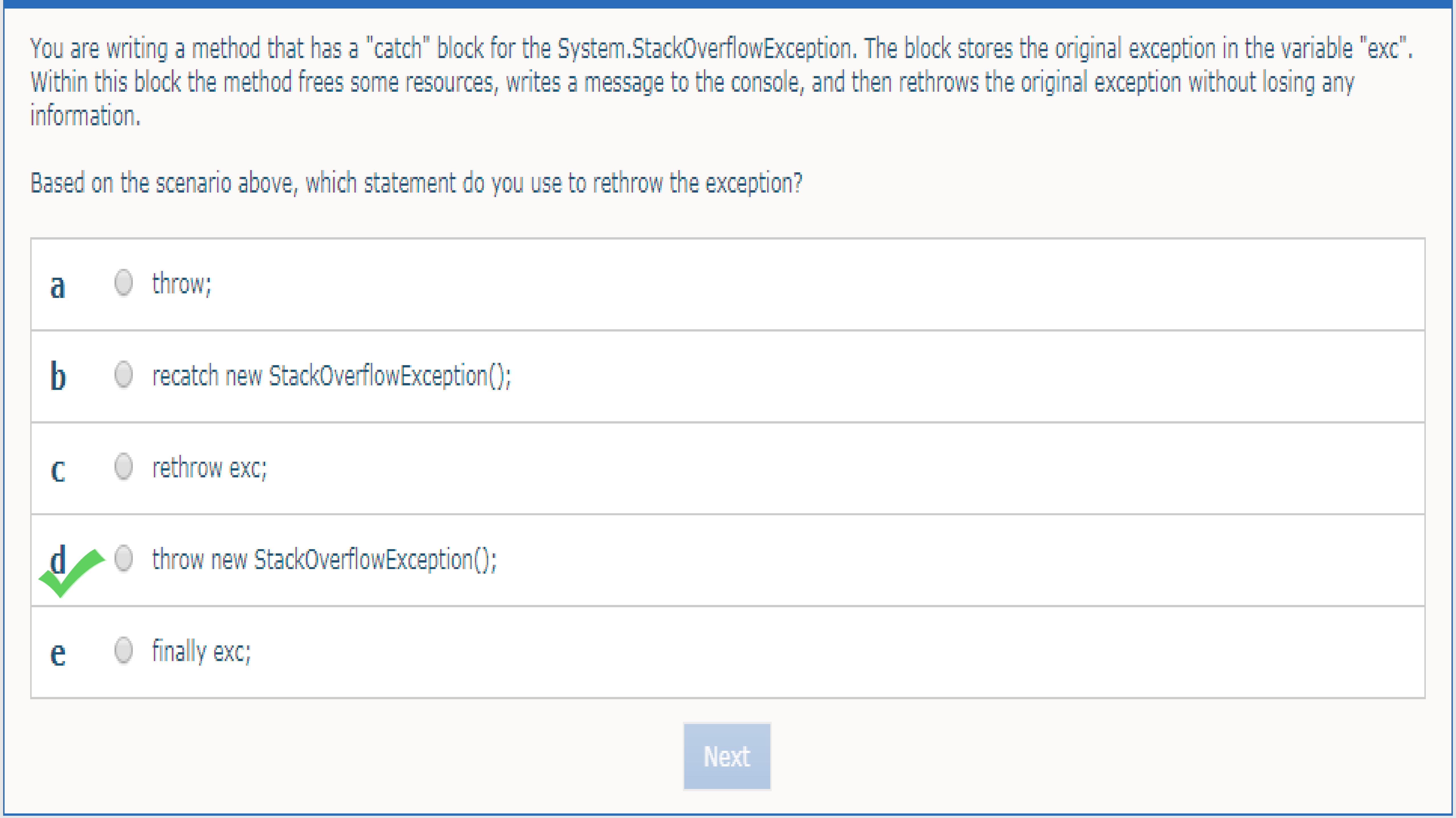
a ○
```
GetX
GetY
False
GetX
False
```

b ○
```
GetX
GetY
True
GetX
GetY
False
```

c ○
```
GetX
GetY
True
GetX
True
```

d ○
```
GetX
GetY
False
GetX
GetY
False
```

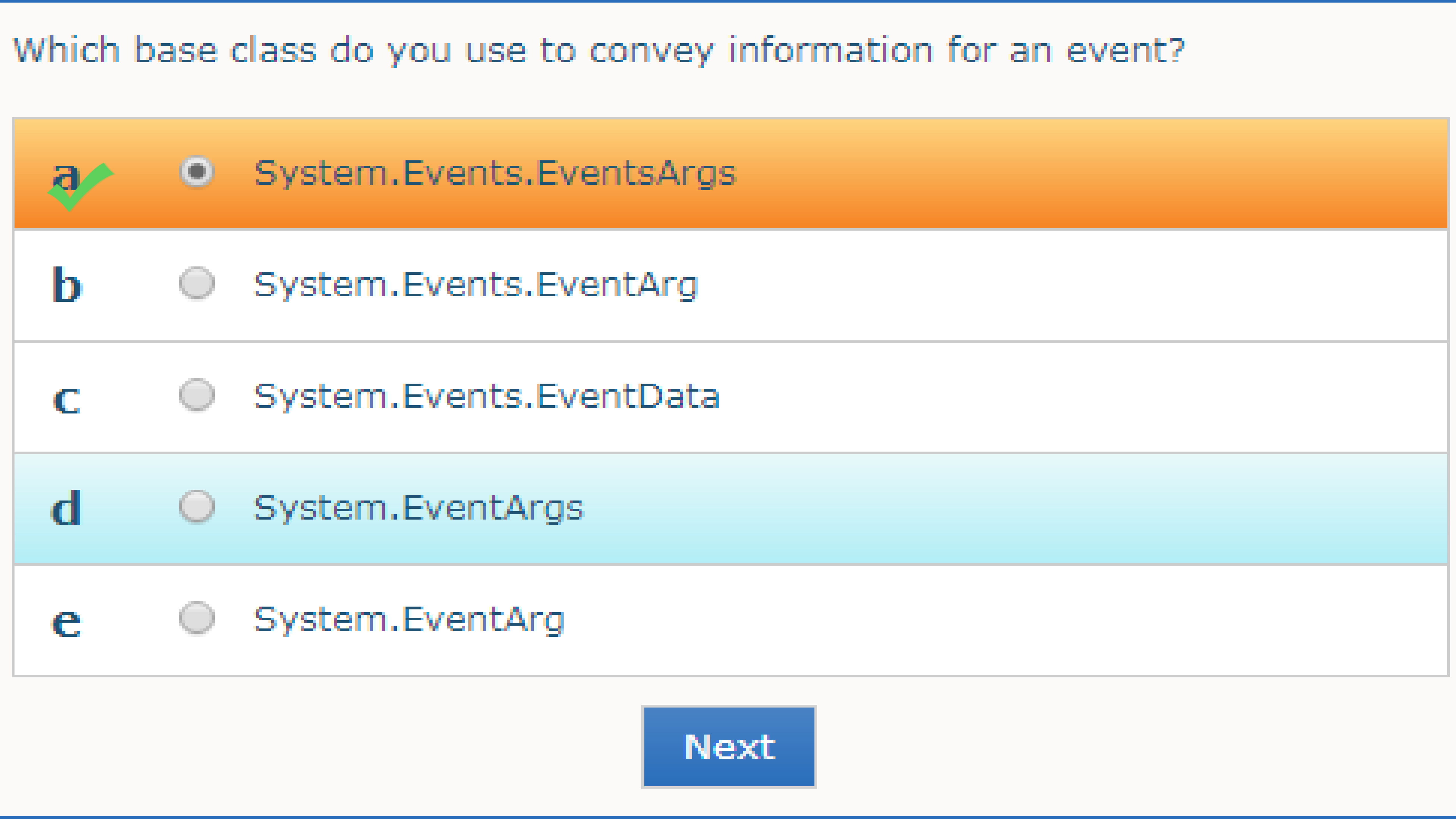
✓ e ○
```
GetX
GetY
True
GetX
GetY
True
```



You are writing a method that has a "catch" block for the System.StackOverflowException. The block stores the original exception in the variable "exc". Within this block the method frees some resources, writes a message to the console, and then rethrows the original exception without losing any information.

Based on the scenario above, which statement do you use to rethrow the exception?

a	○ `throw;`
b	○ `recatch new StackOverflowException();`
c	○ `rethrow exc;`
[✓ d]	○ `throw new StackOverflowException();`
e	○ `finally exc;`



Which base class do you use to convey information for an event?

[✓ a ● System.Events.EventsArgs]

b ○ System.Events.EventArg

c ○ System.Events.EventData

d ○ System.EventArgs

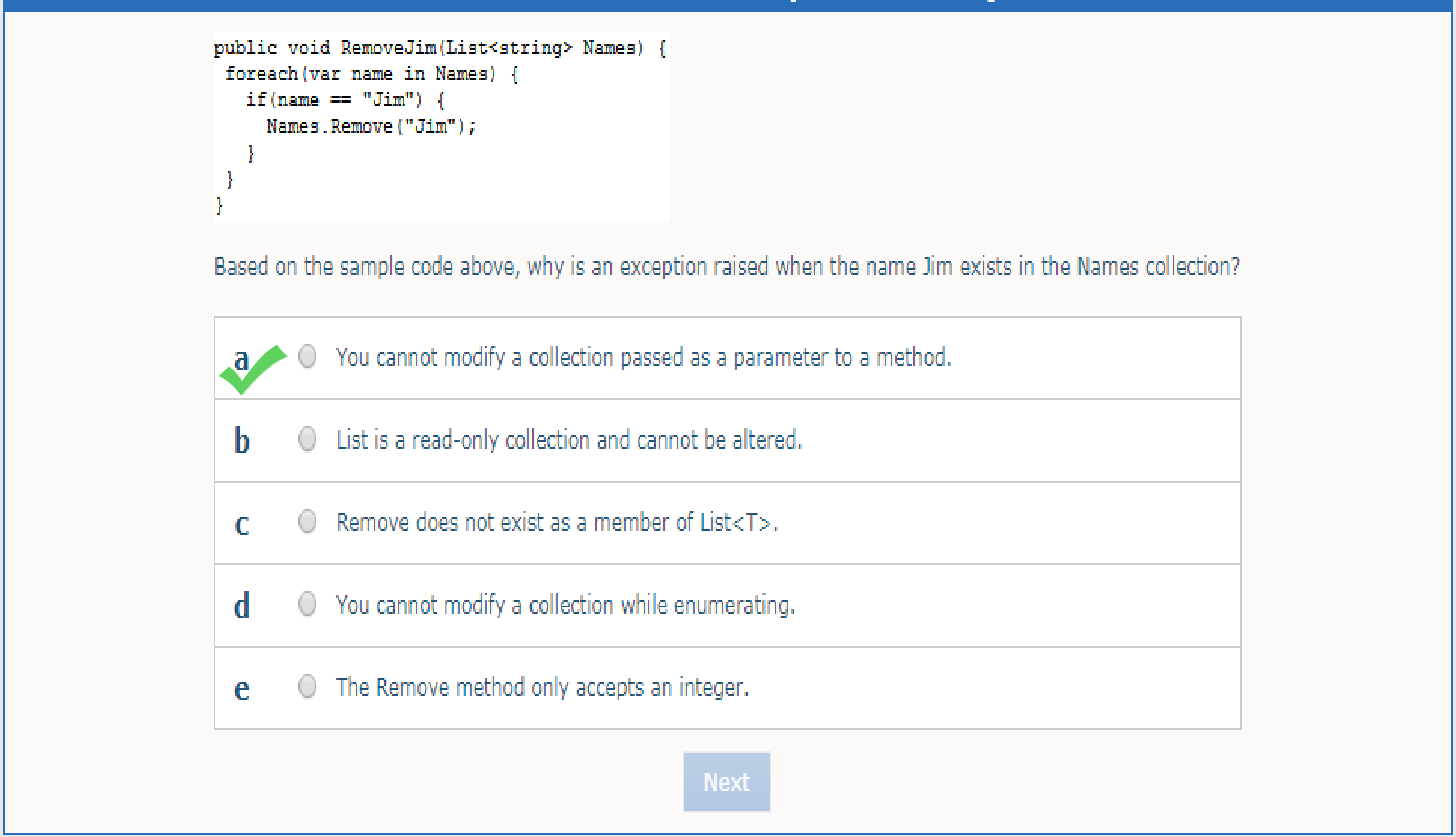
e ○ System.EventArg

Next

```
using System;
using System.Linq;
using System.Collections.Generic;
class Program
{
    static void Main()
    {
        List<Func<int, int>> coinFunctions =
            new List<Func<int, int>>();
        foreach (var coin in new int[]{1,5,10,25})
        {
            coinFunctions.Add((count) => coin * count);
        }
        foreach (var f in coinFunctions)
        {
            Console.WriteLine(f(2));
        }
    }
}
```

What is written to the console when you execute the sample code above?

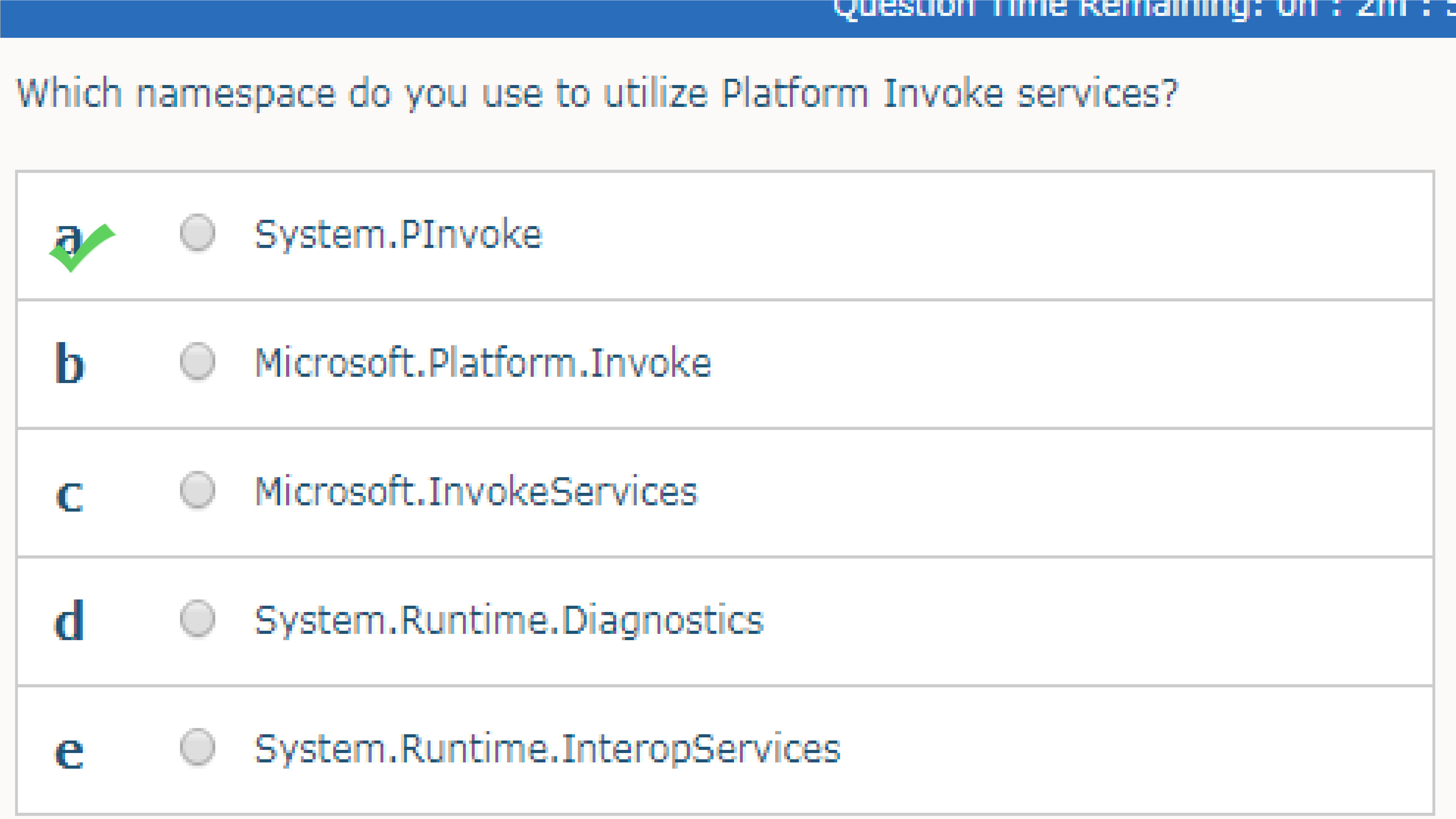
a	○	1 1 1 1
b	○	1 5 10 25
c	○	2 2 2 2
[✓ d]	○	2 10 20 50
e	○	50 50 50



```
public void RemoveJim(List<string> Names) {
 foreach(var name in Names) {
   if(name == "Jim") {
     Names.Remove("Jim");
   }
 }
}
```

Based on the sample code above, why is an exception raised when the name Jim exists in the Names collection?

a ✓	○ You cannot modify a collection passed as a parameter to a method.
b	○ List is a read-only collection and cannot be altered.
c	○ Remove does not exist as a member of List<T>.
d	○ You cannot modify a collection while enumerating.
e	○ The Remove method only accepts an integer.



Question Time Remaining: 0h : 2m :

Which namespace do you use to utilize Platform Invoke services?

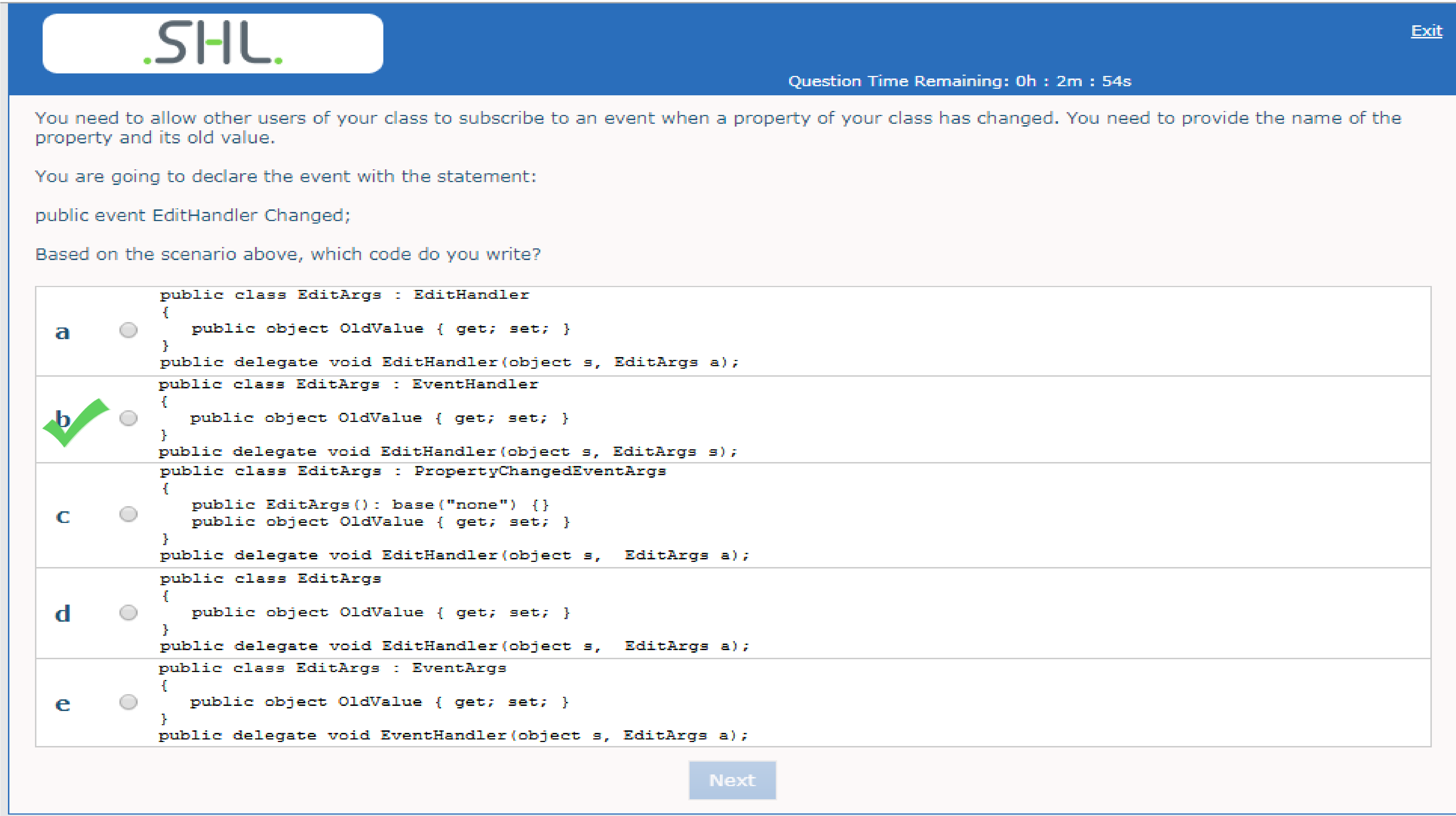
[✓ a] System.PInvoke

b Microsoft.Platform.Invoke

c Microsoft.InvokeServices

d System.Runtime.Diagnostics

e System.Runtime.InteropServices



Question Time Remaining: 0h : 2m : 54s

You need to allow other users of your class to subscribe to an event when a property of your class has changed. You need to provide the name of the property and its old value.

You are going to declare the event with the statement:

public event EditHandler Changed;

Based on the scenario above, which code do you write?

a

```
public class EditArgs : EditHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs a);
```

[✓ b]

```
public class EditArgs : EventHandler
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s, EditArgs s);
```

c

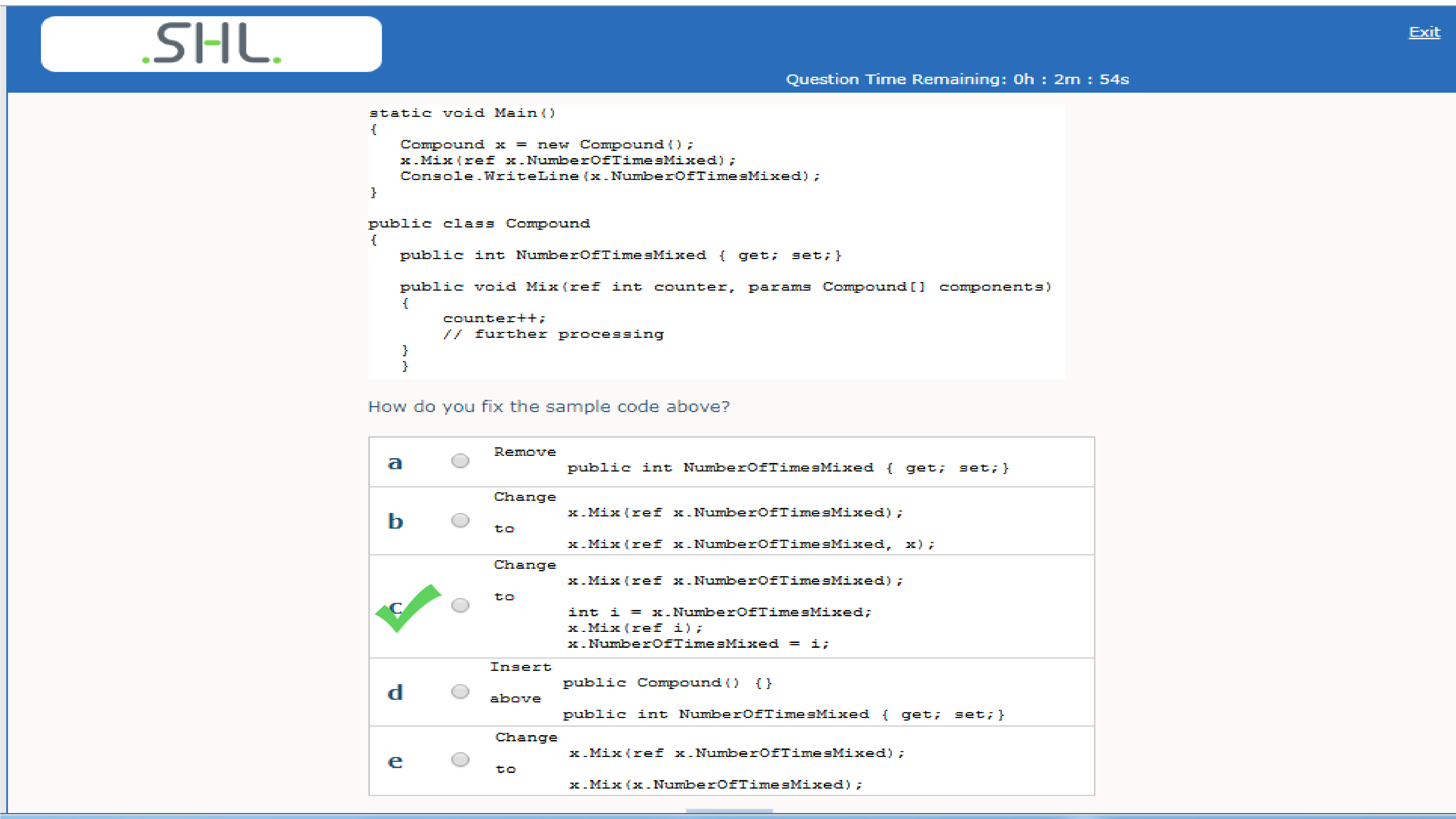
```
public class EditArgs : PropertyChangedEventArgs
{
   public EditArgs(): base("none") {}
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

d

```
public class EditArgs
{
   public object OldValue { get; set; }
}
public delegate void EditHandler(object s,  EditArgs a);
```

e

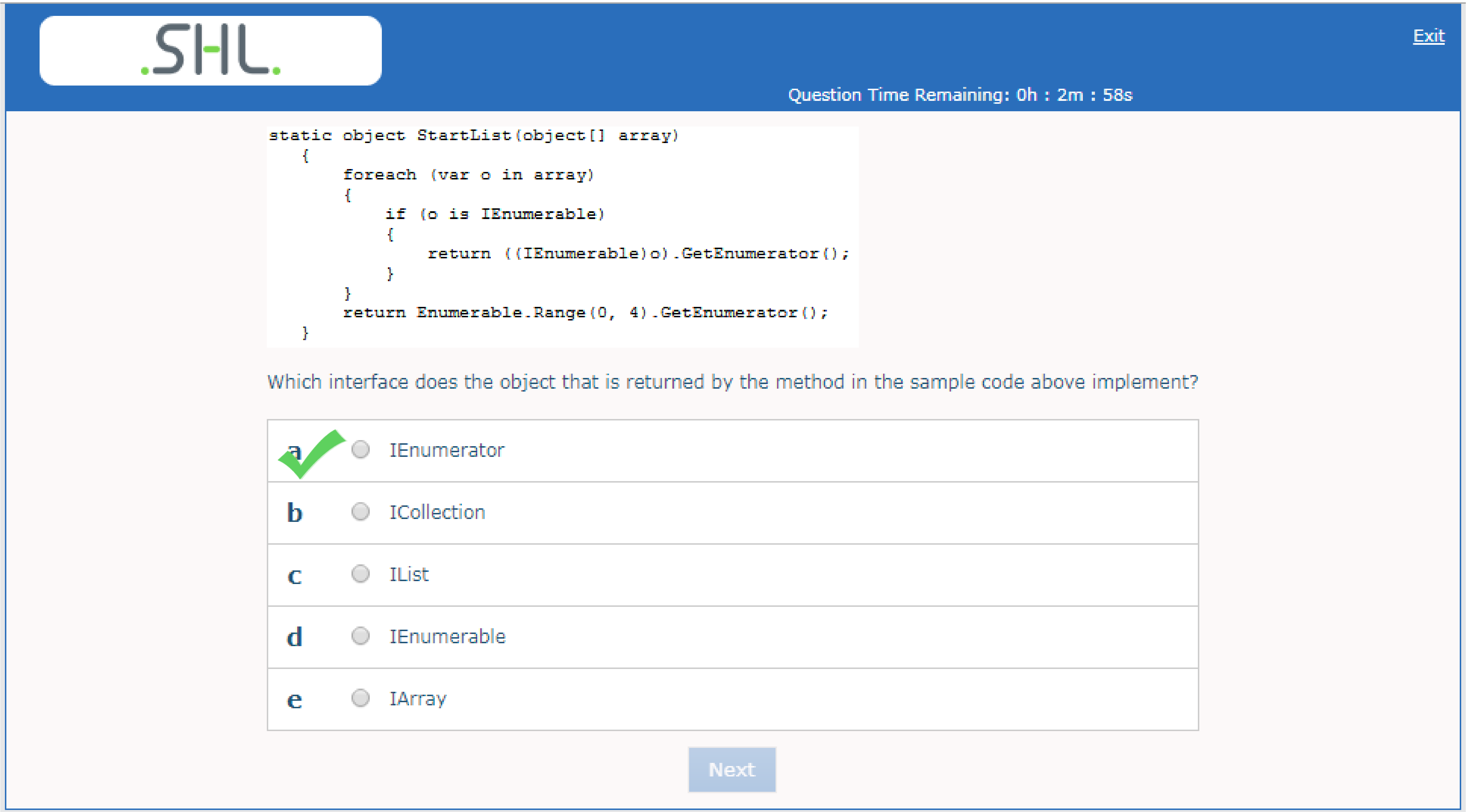
```
public class EditArgs : EventArgs
{
   public object OldValue { get; set; }
}
public delegate void EventHandler(object s, EditArgs a);
```



```
static void Main()
{
   Compound x = new Compound();
   x.Mix(ref x.NumberOfTimesMixed);
   Console.WriteLine(x.NumberOfTimesMixed);
}

public class Compound
{
   public int NumberOfTimesMixed { get; set;}

   public void Mix(ref int counter, params Compound[] components)
   {
       counter++;
       // further processing
   }
   }
```

How do you fix the sample code above?

a	○	Remove `public int NumberOfTimesMixed { get; set;}`
b	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(ref x.NumberOfTimesMixed, x);`
✓ c	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `int i = x.NumberOfTimesMixed;` `x.Mix(ref i);` `x.NumberOfTimesMixed = i;`
d	○	Insert `public Compound() {}` above `public int NumberOfTimesMixed { get; set;}`
e	○	Change `x.Mix(ref x.NumberOfTimesMixed);` to `x.Mix(x.NumberOfTimesMixed);`



Exit

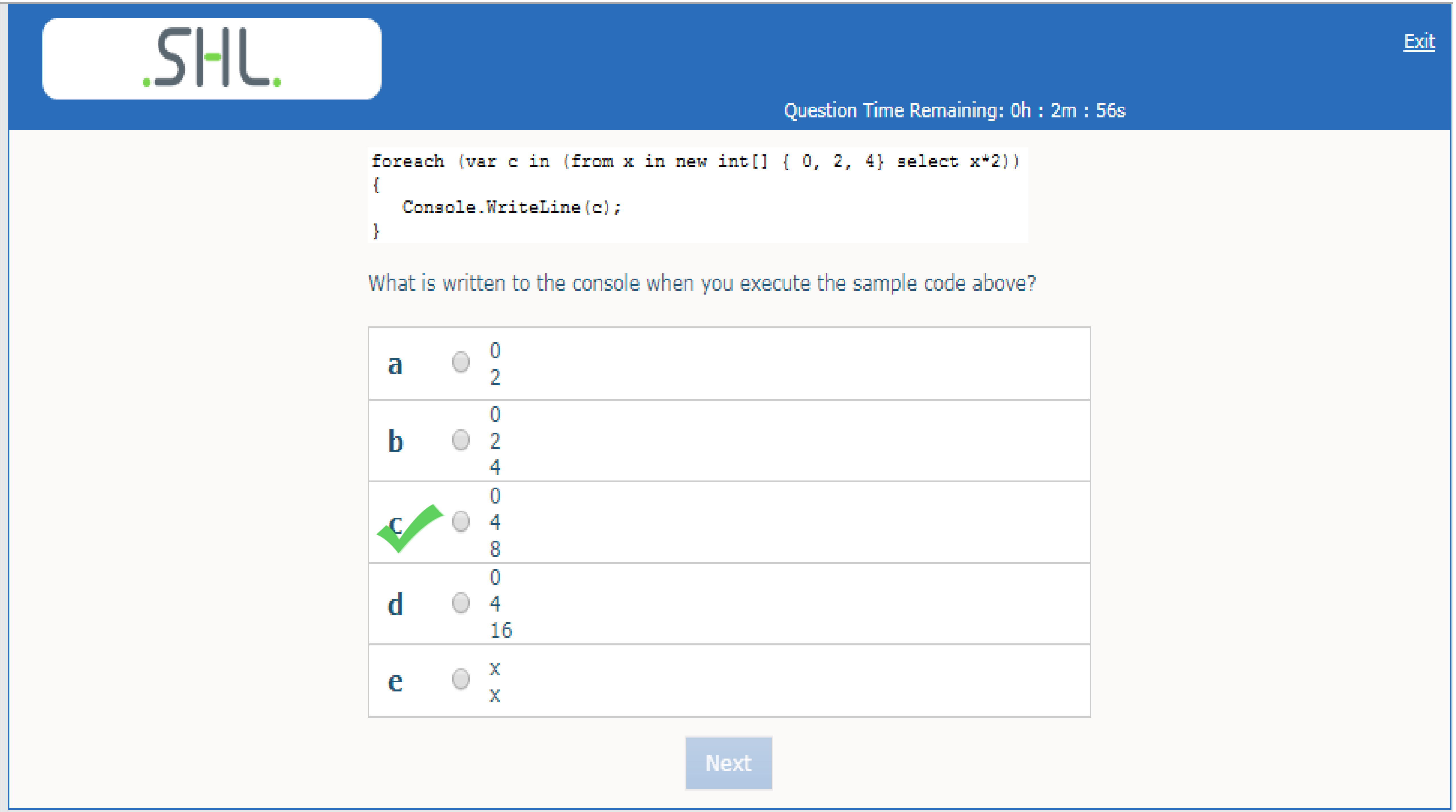
Question Time Remaining: 0h : 2m : 58s

```
static object StartList(object[] array)
   {
        foreach (var o in array)
        {
            if (o is IEnumerable)
            {
                return ((IEnumerable)o).GetEnumerator();
            }
        }
        return Enumerable.Range(0, 4).GetEnumerator();
   }
```

Which interface does the object that is returned by the method in the sample code above implement?

[✓ a]	IEnumerator
b	ICollection
c	IList
d	IEnumerable
e	IArray

Next



```
foreach (var c in (from x in new int[] { 0, 2, 4} select x*2))
{
   Console.WriteLine(c);
}
```

What is written to the console when you execute the sample code above?

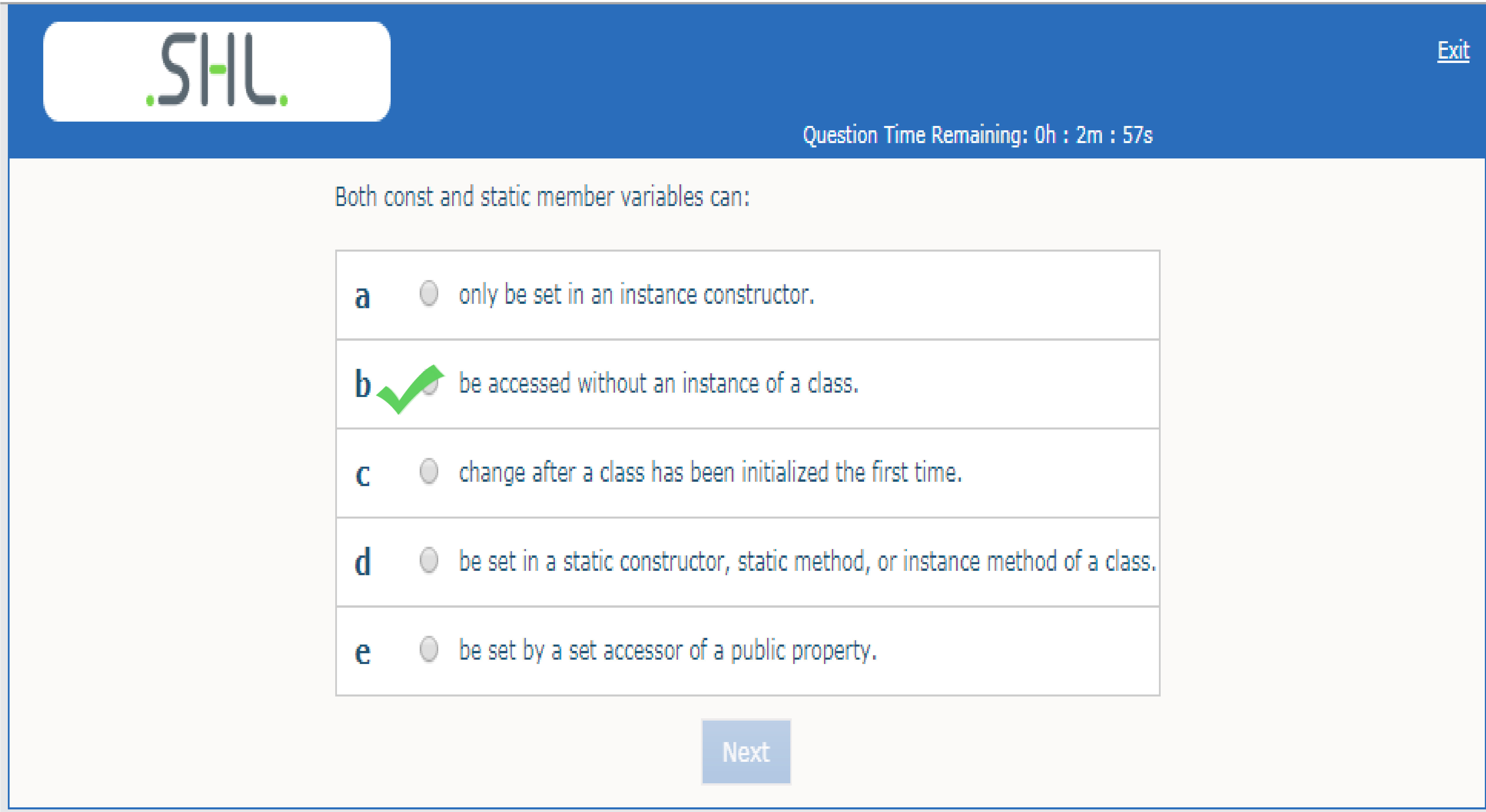
a
0
2

b
0
2
4

[✓ c]
0
4
8

d
0
4
16

e
x
x



Exit

Question Time Remaining: 0h : 2m : 57s

Both const and static member variables can:

a only be set in an instance constructor.

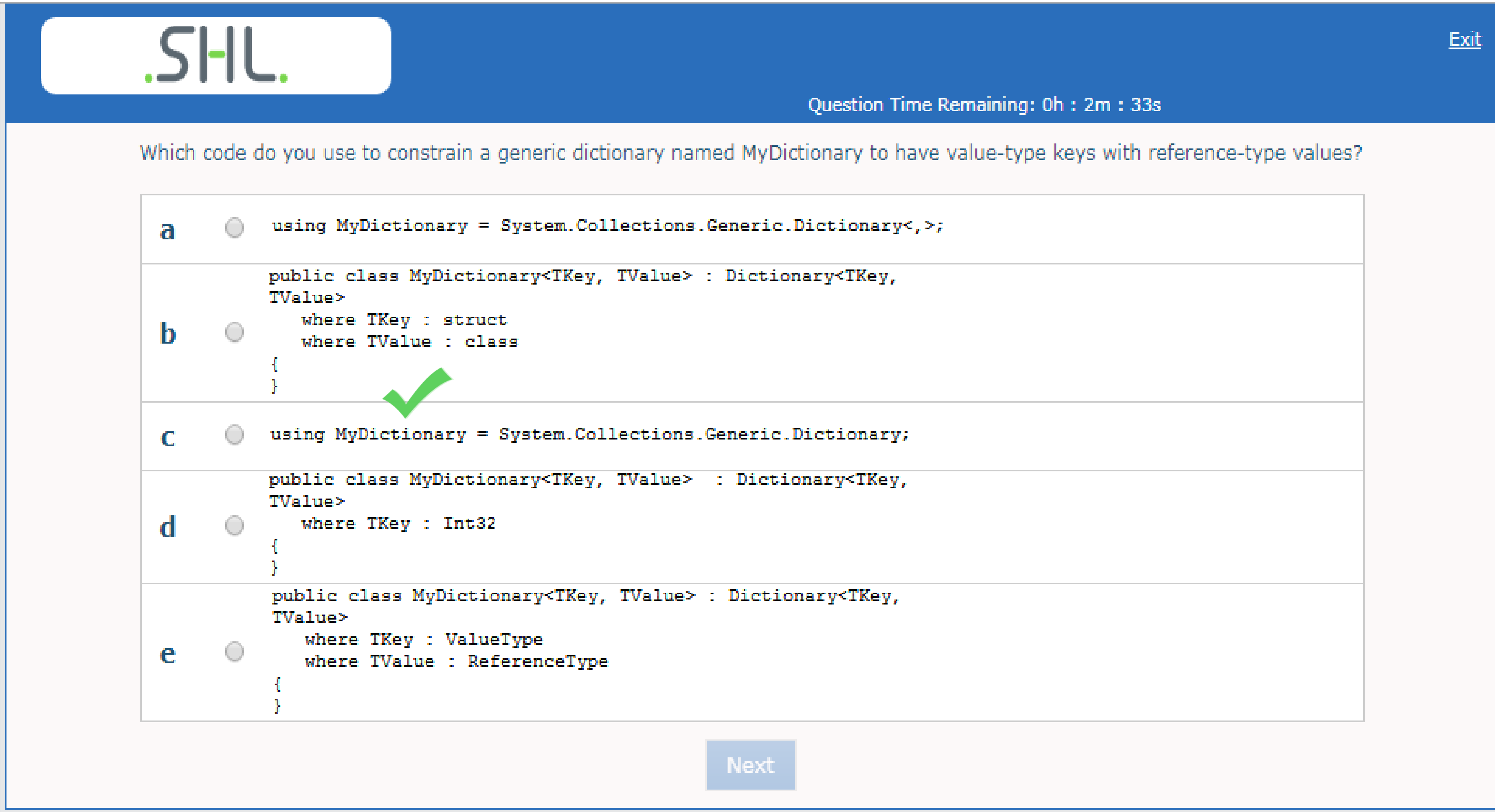
b [✓ be accessed without an instance of a class.]

c change after a class has been initialized the first time.

d be set in a static constructor, static method, or instance method of a class.

e be set by a set accessor of a public property.

Next



Question Time Remaining: 0h : 2m : 33s

Which code do you use to constrain a generic dictionary named MyDictionary to have value-type keys with reference-type values?

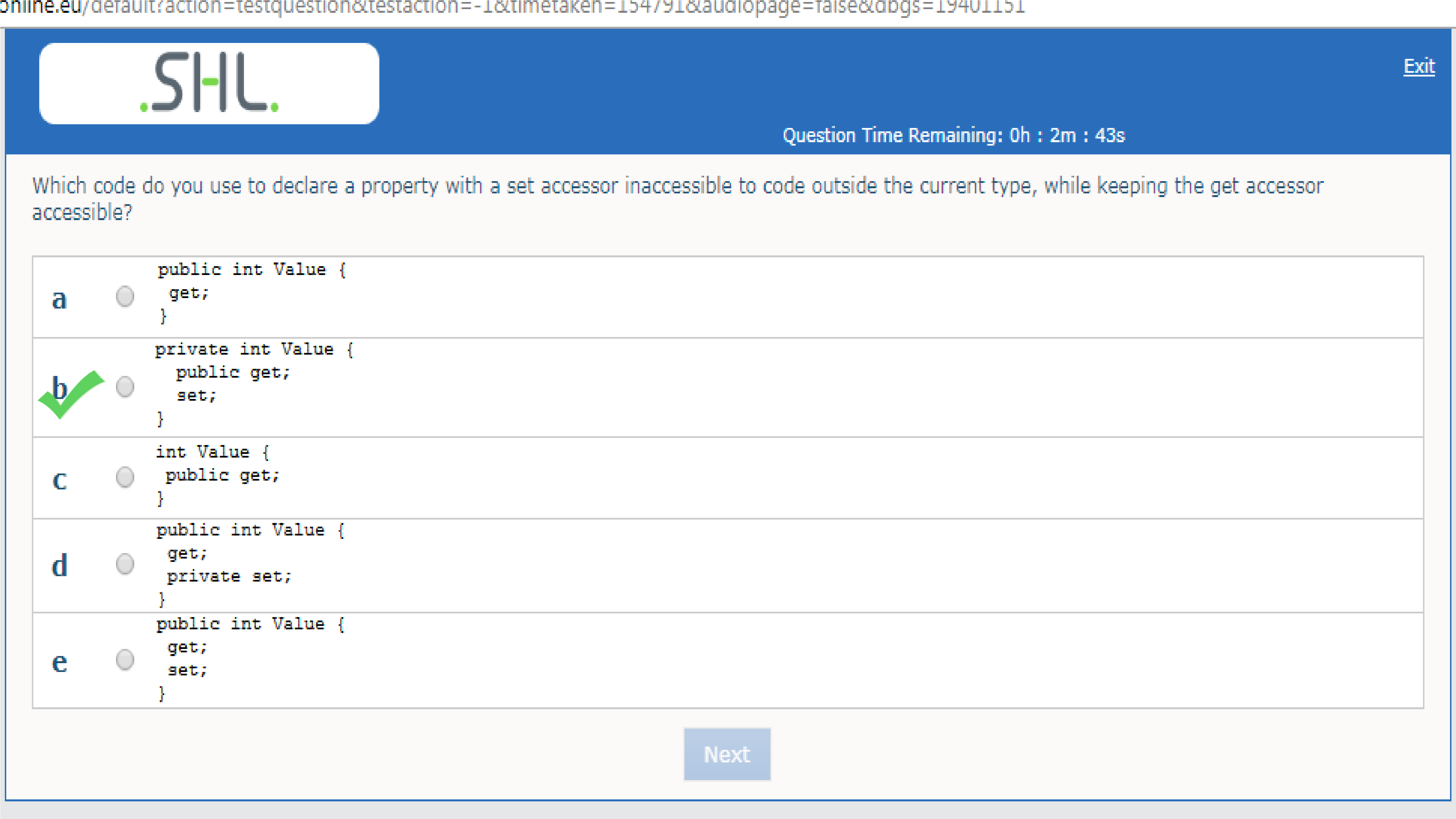
a
```
using MyDictionary = System.Collections.Generic.Dictionary<,>;
```

[✓ b]
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : struct
   where TValue : class
{
}
```

c
```
using MyDictionary = System.Collections.Generic.Dictionary;
```

d
```
public class MyDictionary<TKey, TValue>  : Dictionary<TKey,
TValue>
   where TKey : Int32
{
}
```

e
```
public class MyDictionary<TKey, TValue> : Dictionary<TKey,
TValue>
   where TKey : ValueType
   where TValue : ReferenceType
{
}
```


online.eu/default?action=testquestion&testaction=-1&timetaken=154791&audiopage=false&dbgs=19401151

.SHL.

Exit

Question Time Remaining: 0h : 2m : 43s

Which code do you use to declare a property with a set accessor inaccessible to code outside the current type, while keeping the get accessor accessible?

a
```
public int Value {
 get;
}
```

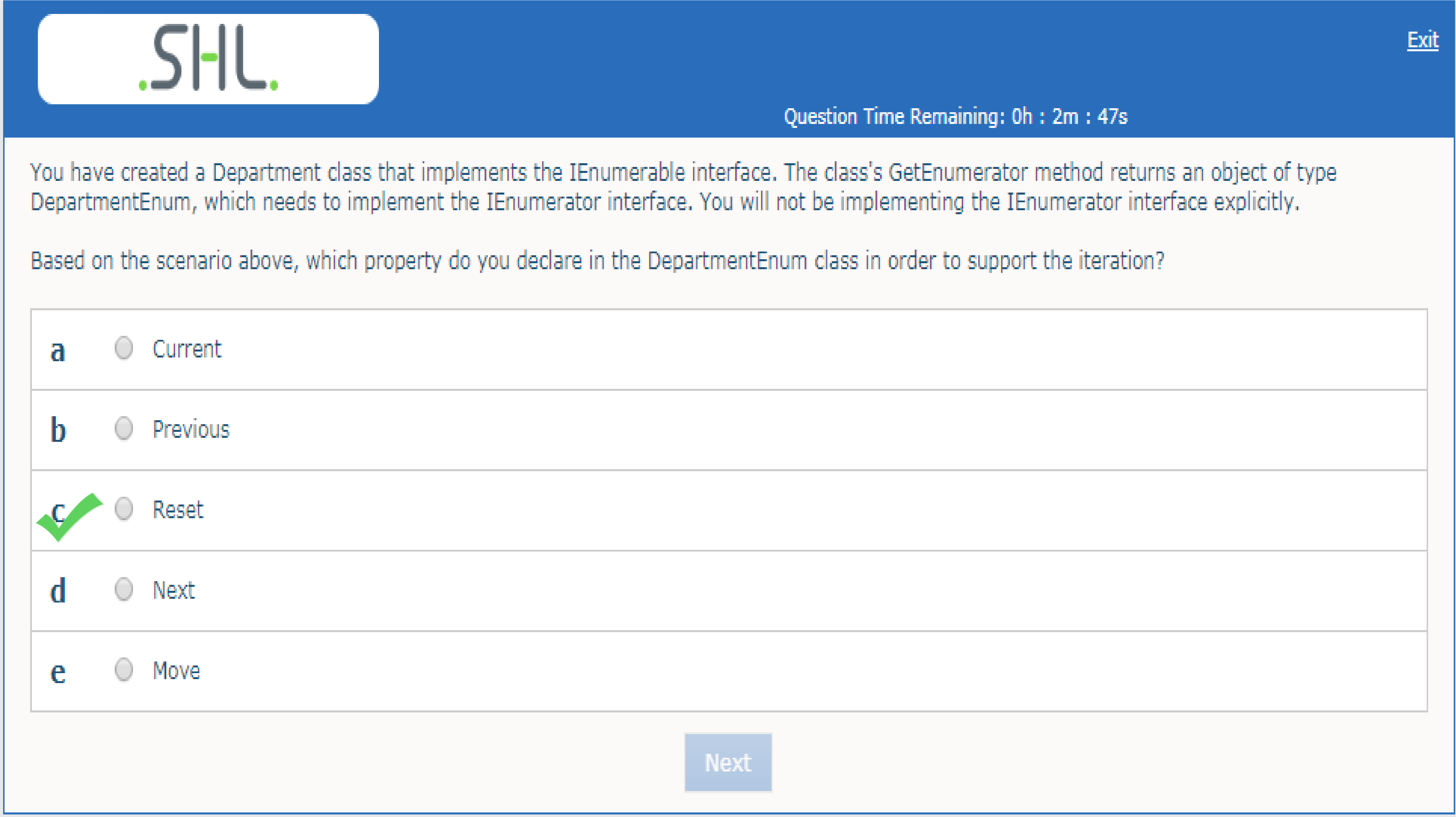
[✓ b]
```
private int Value {
  public get;
  set;
}
```

c
```
int Value {
 public get;
}
```

d
```
public int Value {
 get;
 private set;
}
```

e
```
public int Value {
 get;
 set;
}
```

Next



Exit

Question Time Remaining: 0h : 2m : 47s

You have created a Department class that implements the IEnumerable interface. The class's GetEnumerator method returns an object of type DepartmentEnum, which needs to implement the IEnumerator interface. You will not be implementing the IEnumerator interface explicitly.

Based on the scenario above, which property do you declare in the DepartmentEnum class in order to support the iteration?

a ○ Current

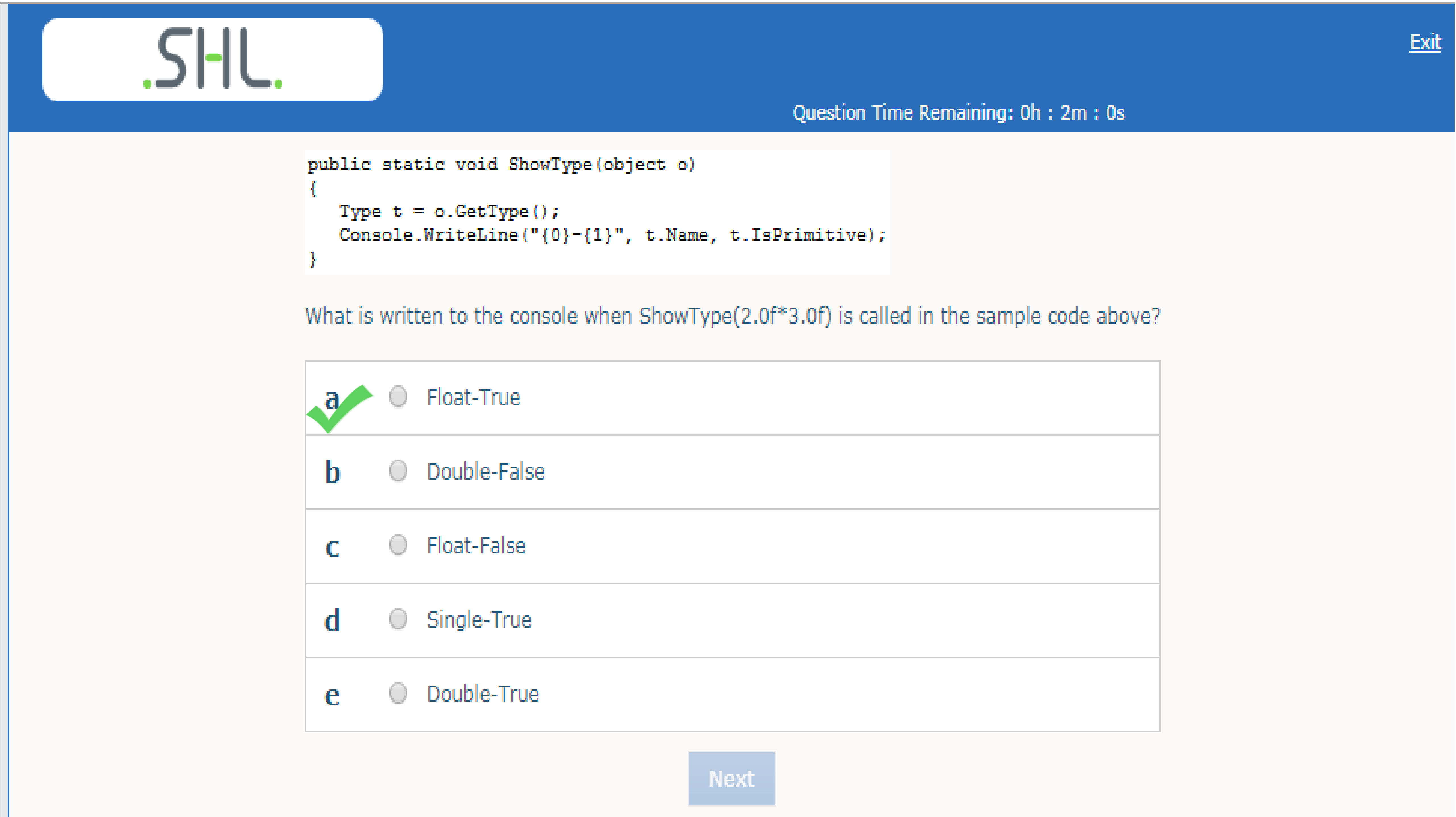
b ○ Previous

[✓ c] ○ Reset

d ○ Next

e ○ Move

Next



Exit

Question Time Remaining: 0h : 2m : 0s

```
public static void ShowType(object o)
{
   Type t = o.GetType();
   Console.WriteLine("{0}-{1}", t.Name, t.IsPrimitive);
}
```

What is written to the console when ShowType(2.0f*3.0f) is called in the sample code above?

[✓ a] Float-True

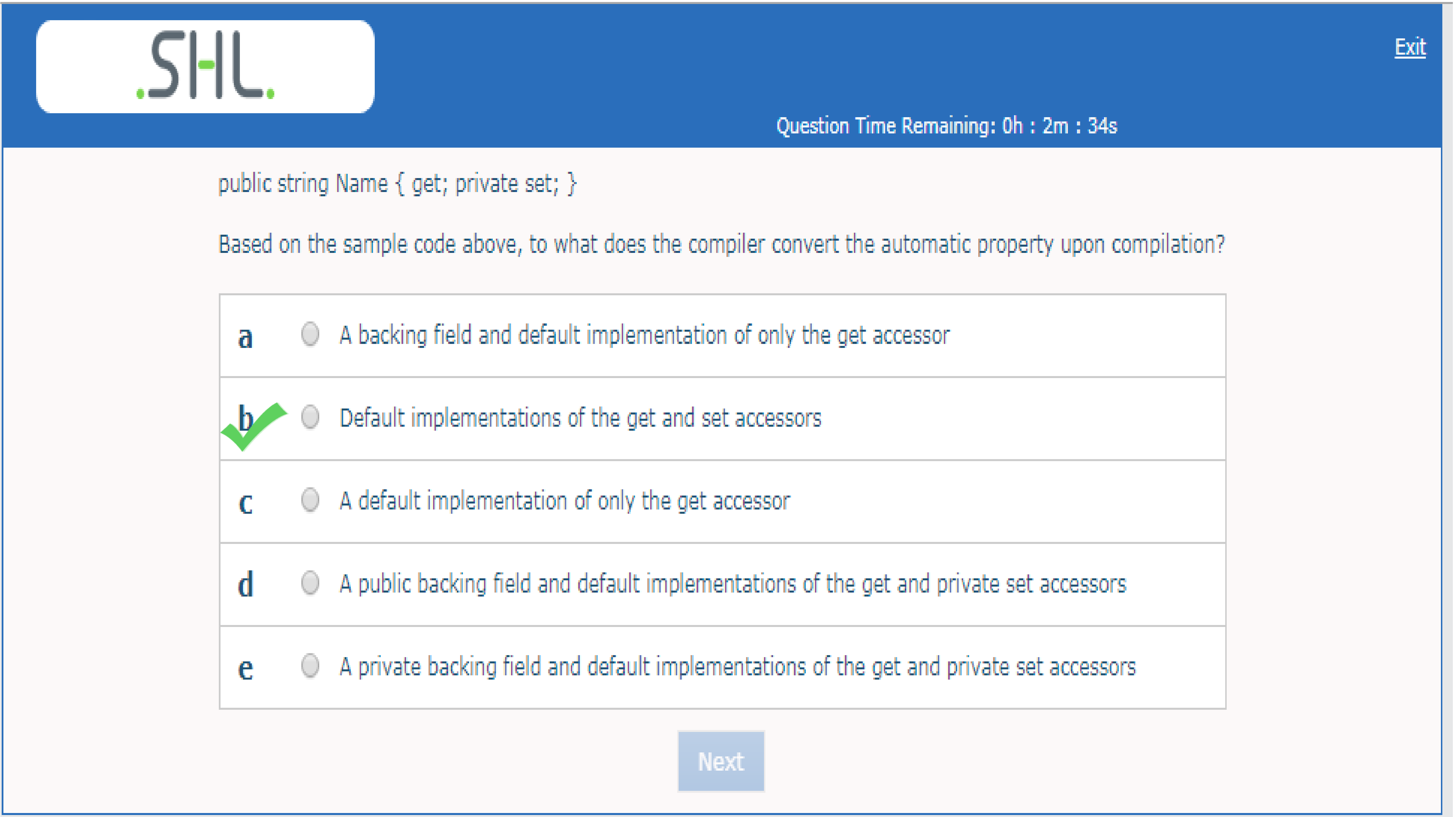
b Double-False

c Float-False

d Single-True

e Double-True

Next



```
public string Name { get; private set; }
```

Based on the sample code above, to what does the compiler convert the automatic property upon compilation?

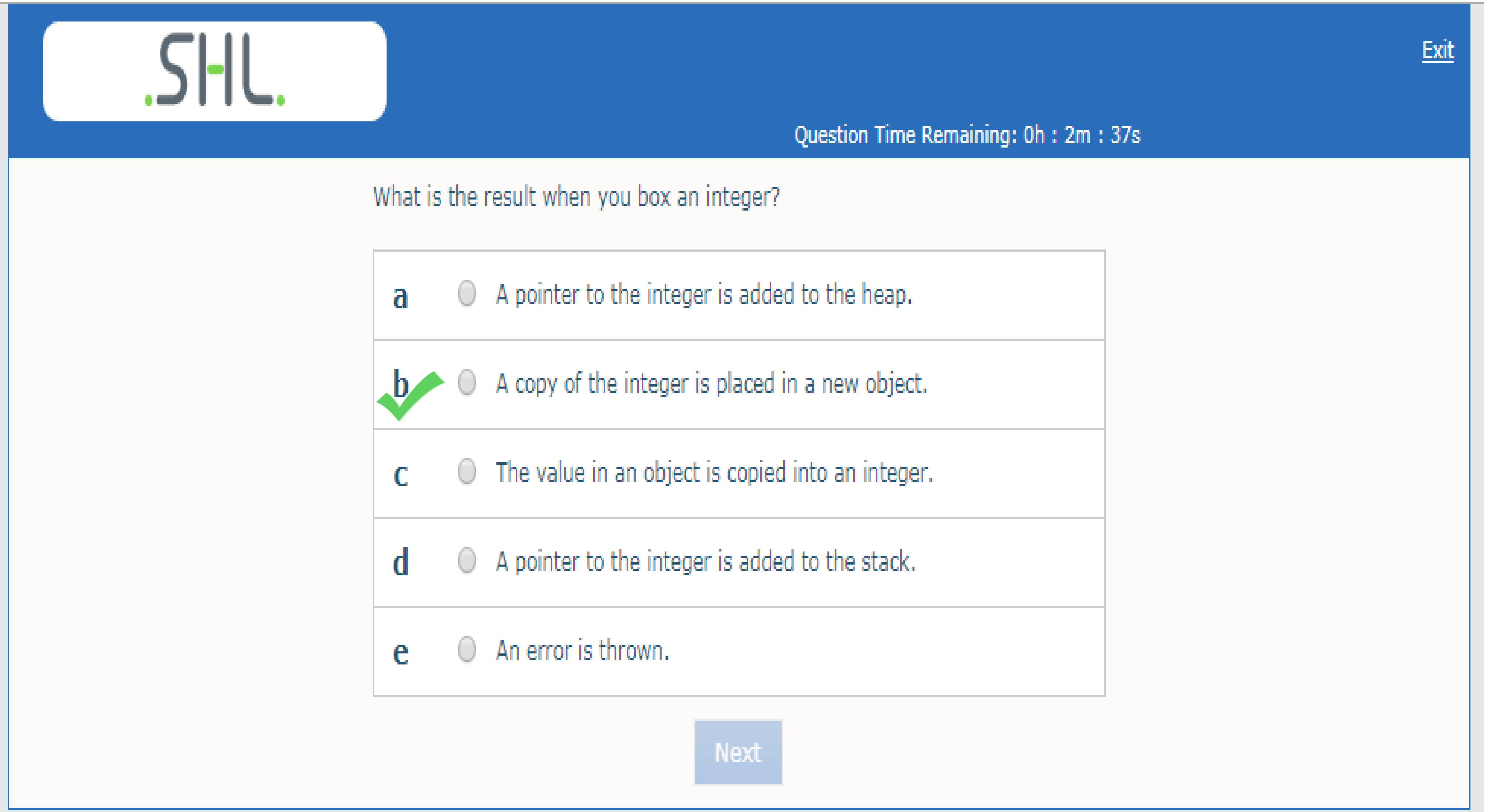
a ○ A backing field and default implementation of only the get accessor

b ✓ ○ Default implementations of the get and set accessors

c ○ A default implementation of only the get accessor

d ○ A public backing field and default implementations of the get and private set accessors

e ○ A private backing field and default implementations of the get and private set accessors

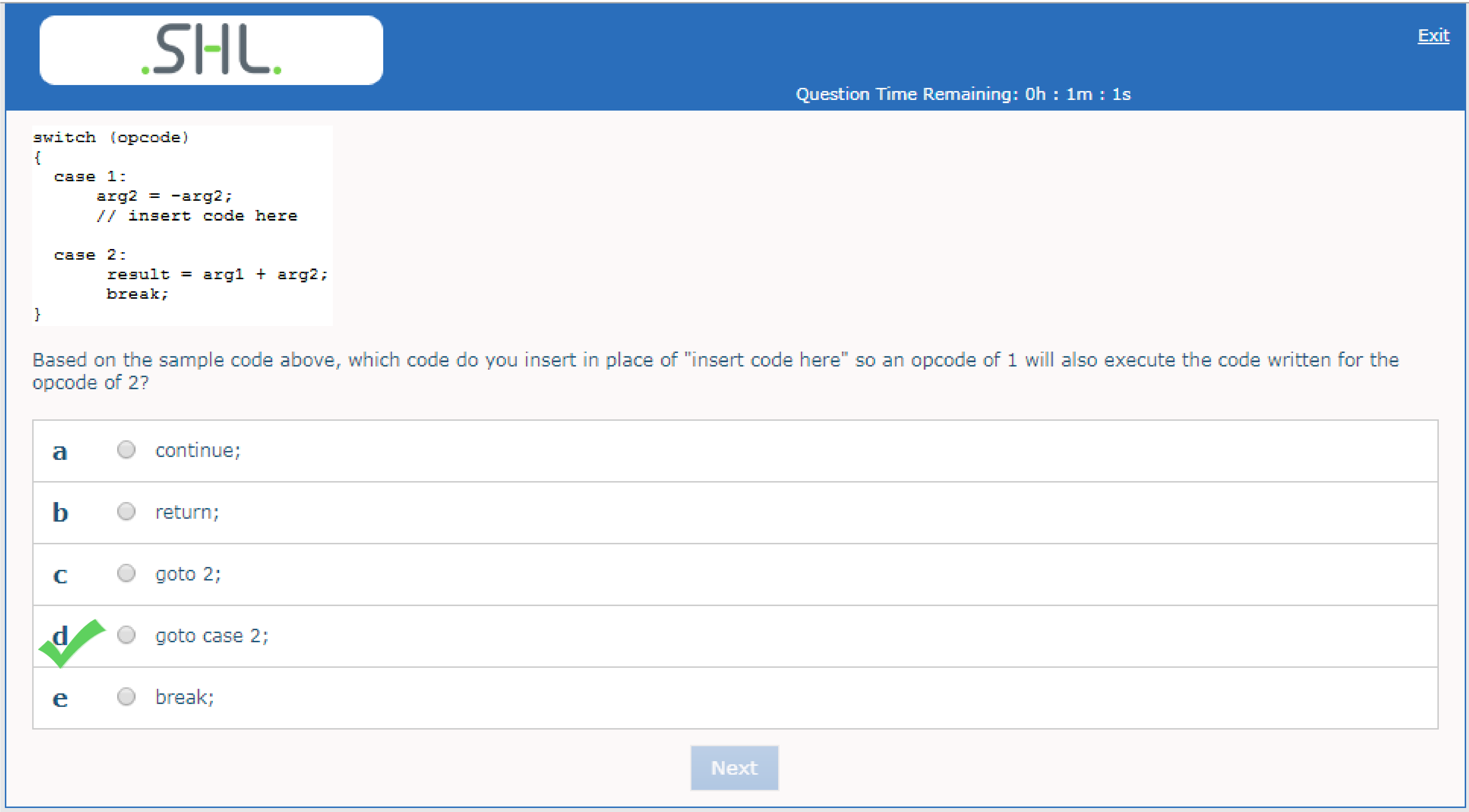


Exit

Question Time Remaining: 0h : 2m : 37s

What is the result when you box an integer?

a	A pointer to the integer is added to the heap.
[✓ b]	A copy of the integer is placed in a new object.
c	The value in an object is copied into an integer.
d	A pointer to the integer is added to the stack.
e	An error is thrown.



Question Time Remaining: 0h : 1m : 1s

```
switch (opcode)
{
  case 1:
      arg2 = -arg2;
      // insert code here

  case 2:
       result = arg1 + arg2;
       break;
}
```

Based on the sample code above, which code do you insert in place of "insert code here" so an opcode of 1 will also execute the code written for the opcode of 2?

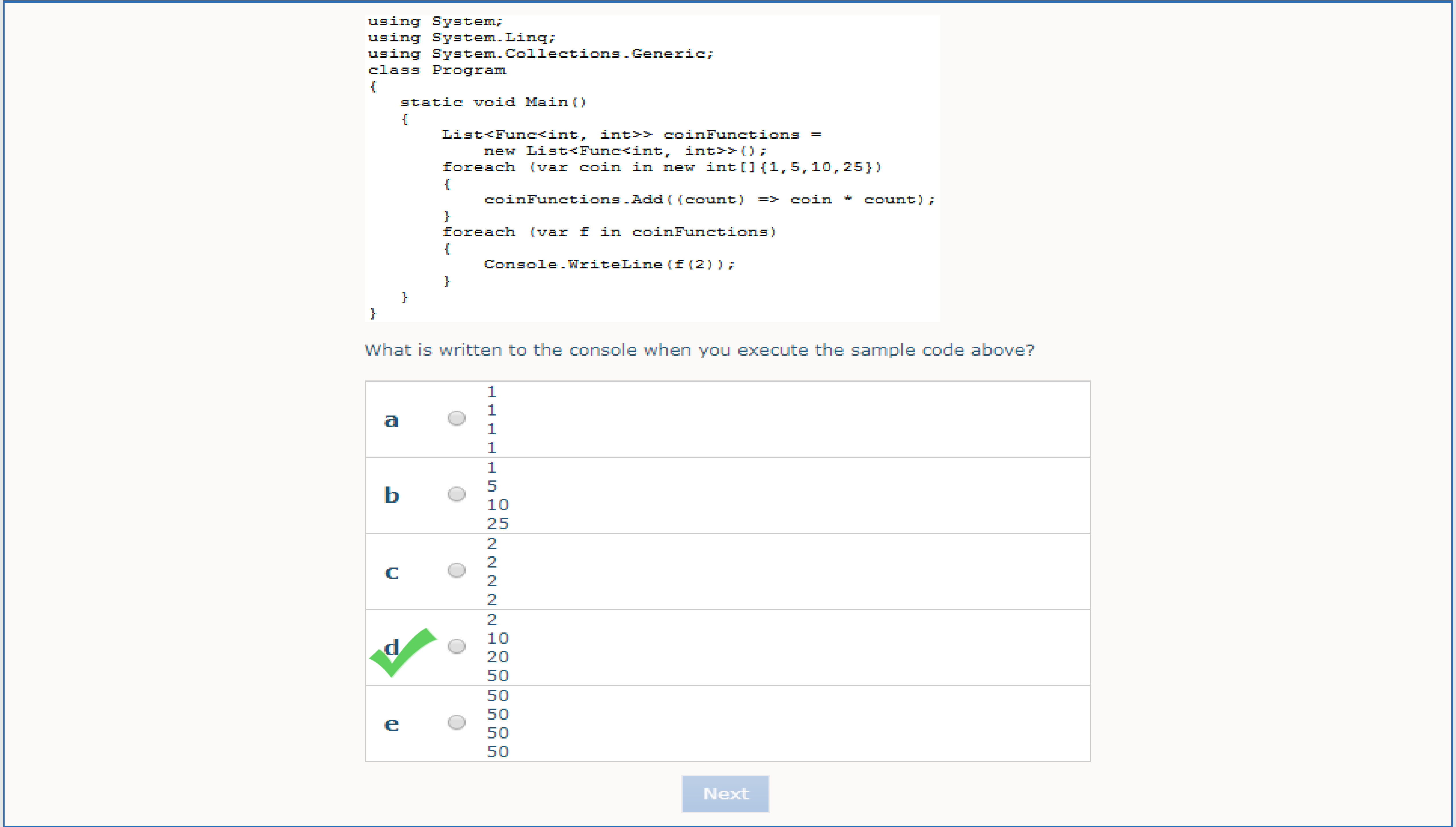
a continue;

b return;

c goto 2;

[✓ d goto case 2;]

e break;

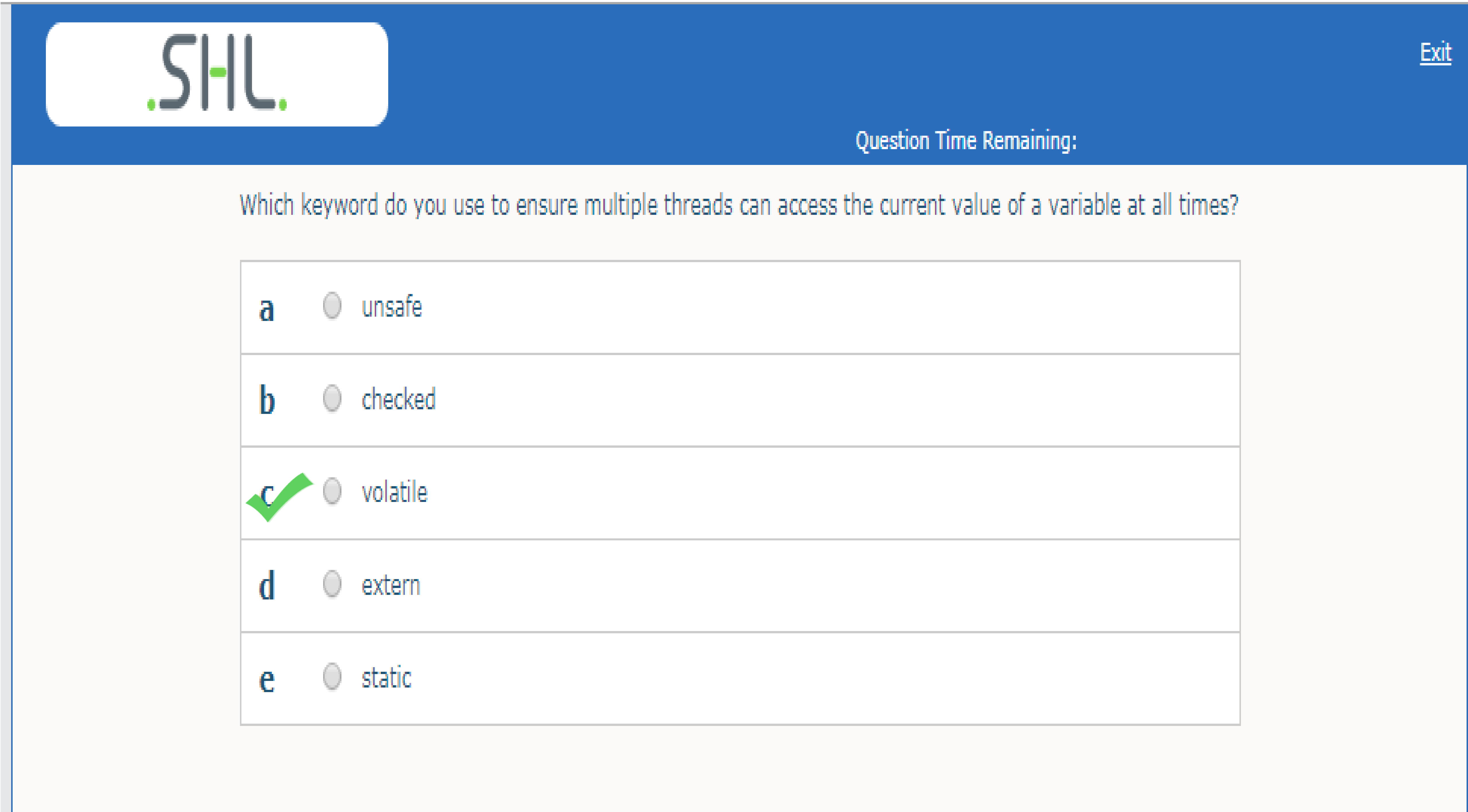


```
using System;
using System.Linq;
using System.Collections.Generic;
class Program
{
    static void Main()
    {
        List<Func<int, int>> coinFunctions =
            new List<Func<int, int>>();
        foreach (var coin in new int[]{1,5,10,25})
        {
            coinFunctions.Add((count) => coin * count);
        }
        foreach (var f in coinFunctions)
        {
            Console.WriteLine(f(2));
        }
    }
}
```

What is written to the console when you execute the sample code above?

a	1 1 1 1
b	1 5 10 25
c	2 2 2 2
[✓ d]	2 10 20 50
e	50 50 50 50

Next



Exit

Question Time Remaining:

Which keyword do you use to ensure multiple threads can access the current value of a variable at all times?

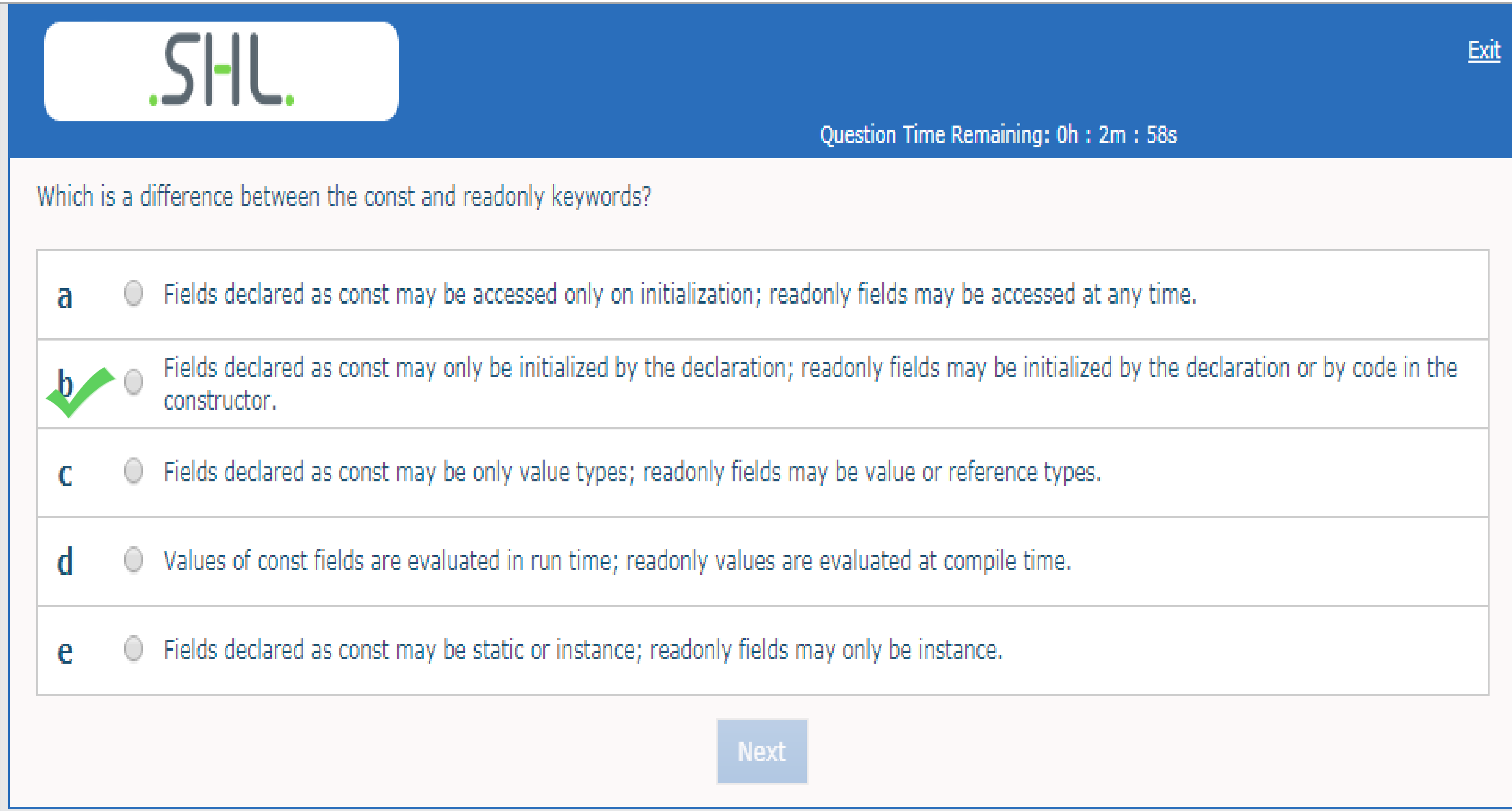
a ○ unsafe

b ○ checked

[✓ c ○ volatile]

d ○ extern

e ○ static



Exit

Question Time Remaining: 0h : 2m : 58s

Which is a difference between the const and readonly keywords?

a Fields declared as const may be accessed only on initialization; readonly fields may be accessed at any time.

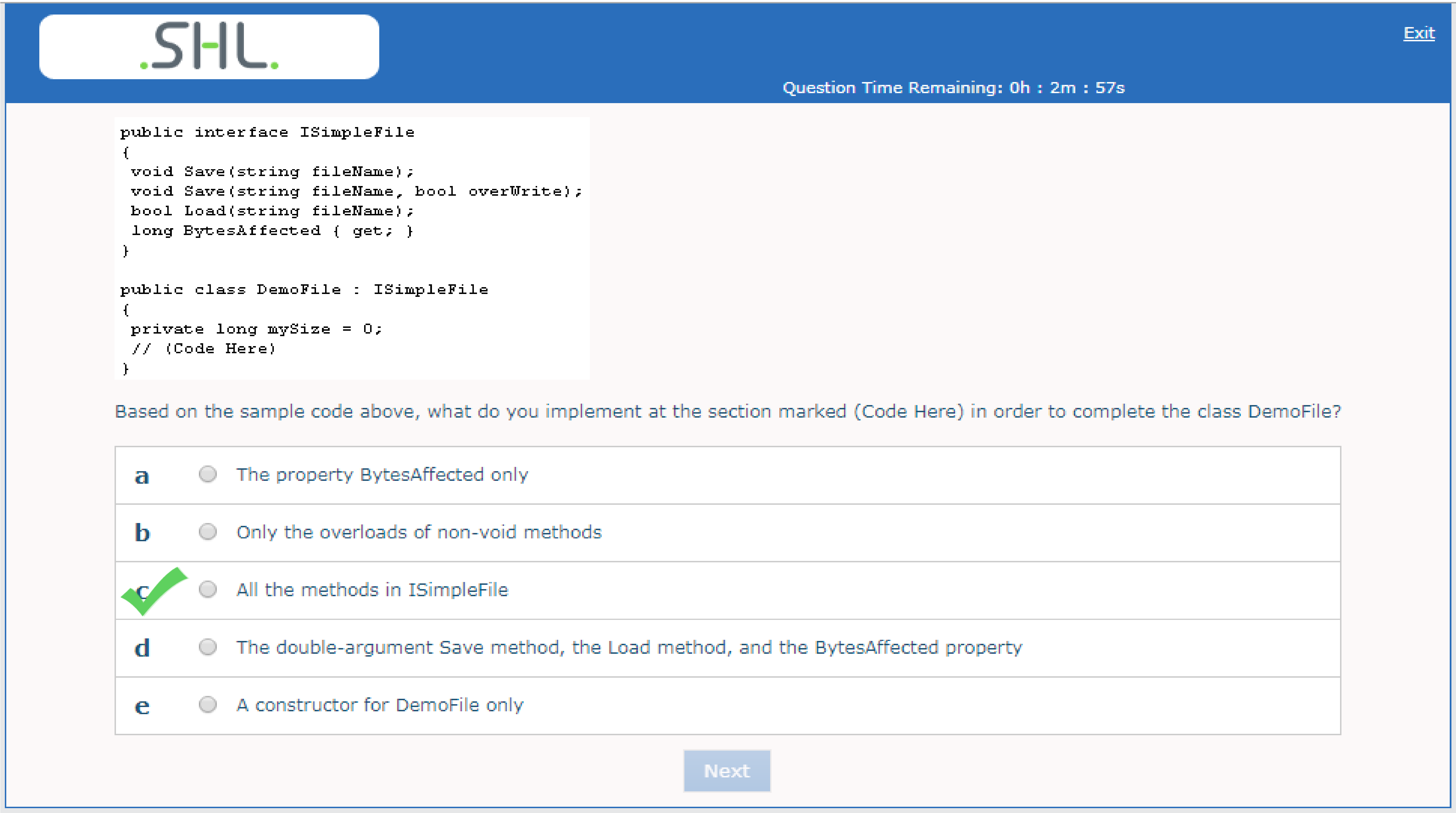
[✓ b] Fields declared as const may only be initialized by the declaration; readonly fields may be initialized by the declaration or by code in the constructor.

c Fields declared as const may be only value types; readonly fields may be value or reference types.

d Values of const fields are evaluated in run time; readonly values are evaluated at compile time.

e Fields declared as const may be static or instance; readonly fields may only be instance.

Next

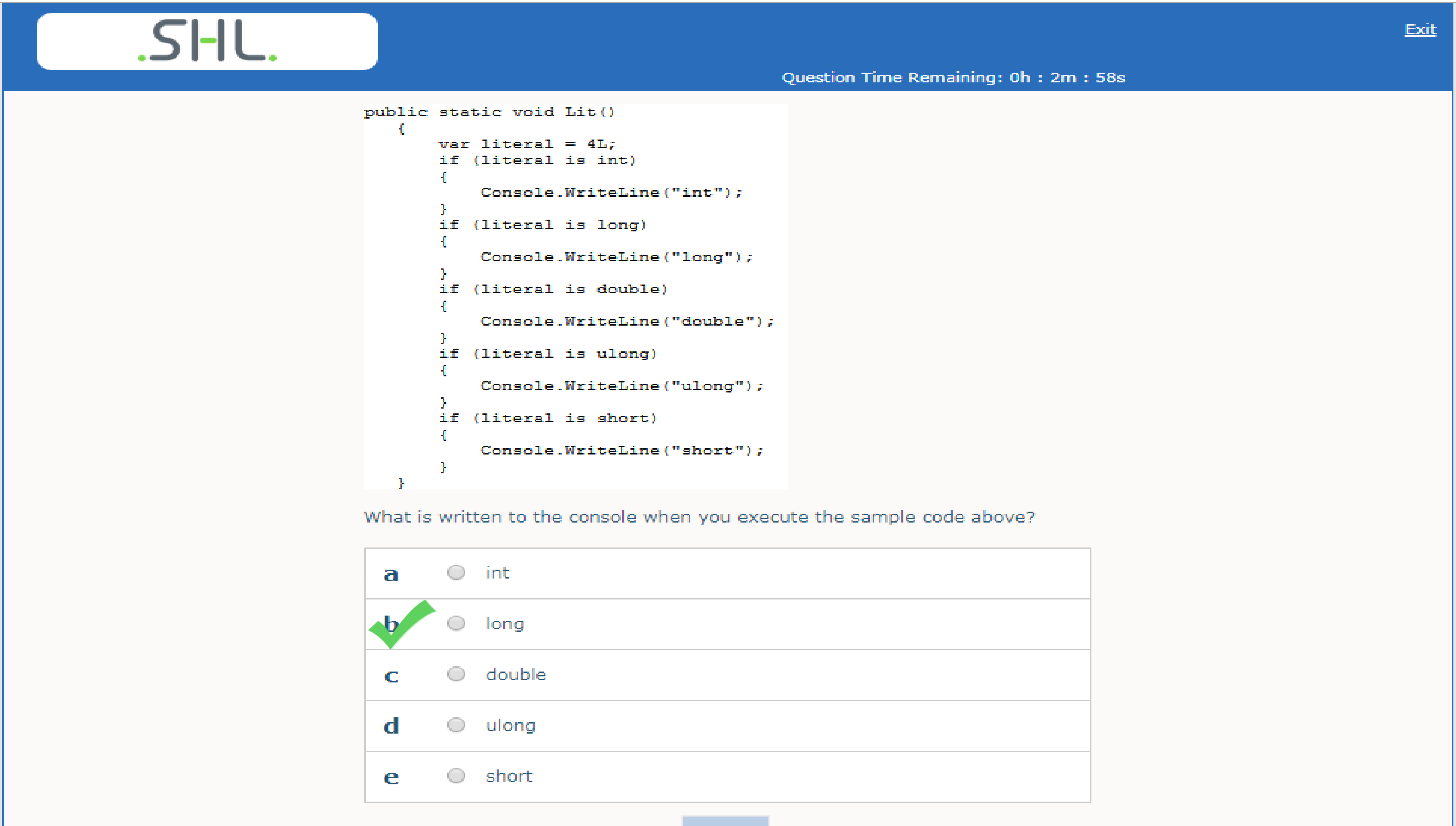


Question Time Remaining: 0h : 2m : 57s

```
public interface ISimpleFile
{
 void Save(string fileName);
 void Save(string fileName, bool overWrite);
 bool Load(string fileName);
 long BytesAffected { get; }
}

public class DemoFile : ISimpleFile
{
 private long mySize = 0;
 // (Code Here)
}
```

Based on the sample code above, what do you implement at the section marked (Code Here) in order to complete the class DemoFile?

a	The property BytesAffected only
b	Only the overloads of non-void methods
[✓ c]	All the methods in ISimpleFile
d	The double-argument Save method, the Load method, and the BytesAffected property
e	A constructor for DemoFile only

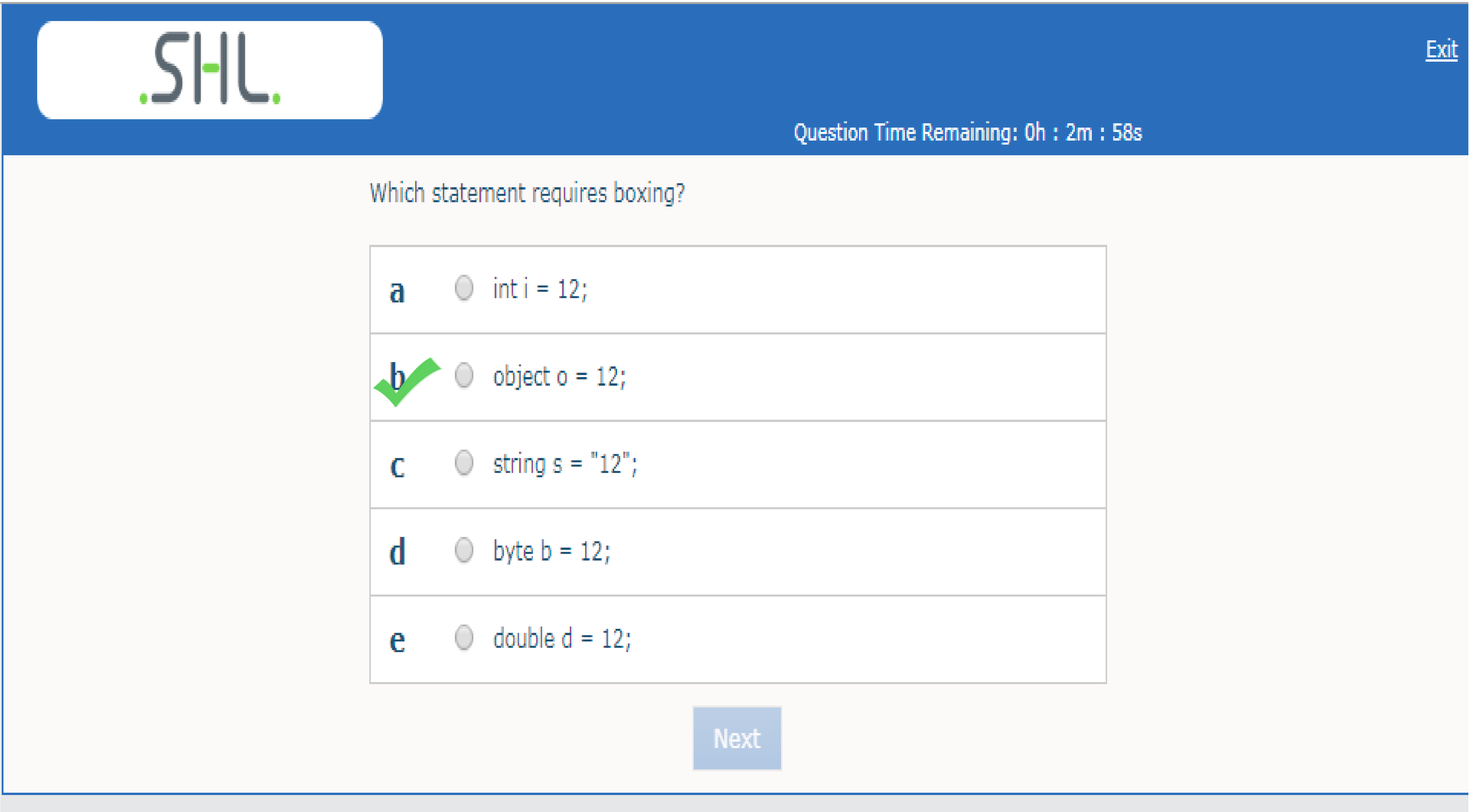


Question Time Remaining: 0h : 2m : 58s

```
public static void Lit()
    {
         var literal = 4L;
         if (literal is int)
         {
              Console.WriteLine("int");
         }
         if (literal is long)
         {
              Console.WriteLine("long");
         }
         if (literal is double)
         {
              Console.WriteLine("double");
         }
         if (literal is ulong)
         {
              Console.WriteLine("ulong");
         }
         if (literal is short)
         {
              Console.WriteLine("short");
         }
    }
```

What is written to the console when you execute the sample code above?

a	int
[✓ b]	long
c	double
d	ulong
e	short



Exit

Question Time Remaining: 0h : 2m : 58s

Which statement requires boxing?

a `int i = 12;`

[✓ b] `object o = 12;`

c `string s = "12";`

d `byte b = 12;`

e `double d = 12;`

Next

Question Time Remaining: 0h : 2m : 51s

```
using System;

class Program {
    static void Main(string[] args) {
         try {
              Console.WriteLine("Level 1");
              try {
                   Console.WriteLine("Level 2");
                   throw new Exception();
                   goto exit;
              } catch {
              } finally {
                   Console.WriteLine("Level 2 Finished");
              }
         } finally {
              Console.WriteLine("Level 1 Finished");
         }
    exit: ;
    }
}
```

Based on the sample code above, what is written to the console?

[✓ a]
Level 1
Level 2
Level 2 Finished
Level 1 Finished

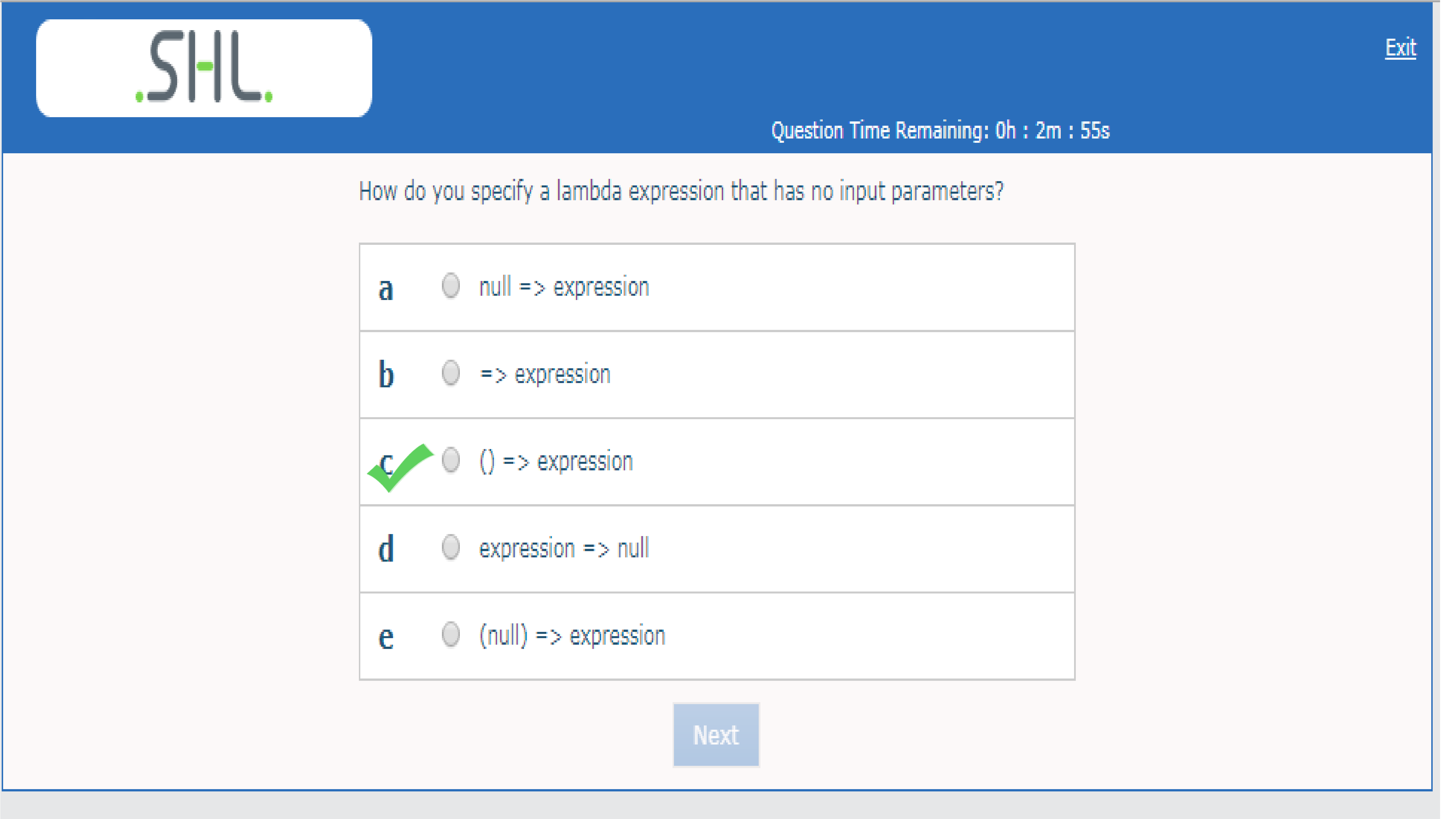
b
Level 1
Level 2

c
Level 1
Level 2
Level 1 Finished

d
Level 1
Level 2
Level 2 Finished

e
Level 1
Level 2
Level 1 Finished
Level 2 Finished

Next



Exit

Question Time Remaining: 0h : 2m : 55s

How do you specify a lambda expression that has no input parameters?

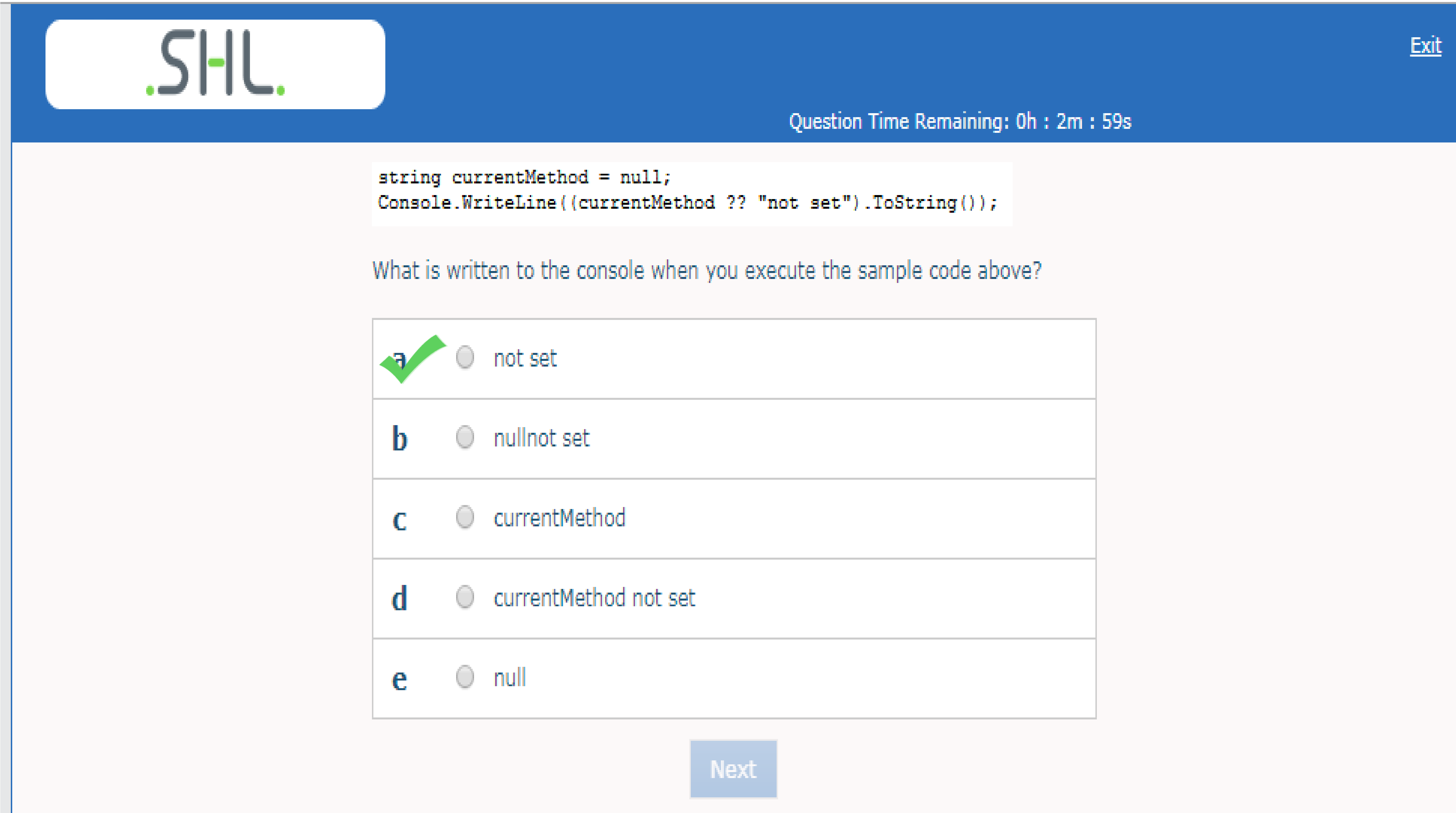
a null => expression

b => expression

[✓ c () => expression]

d expression => null

e (null) => expression



Exit

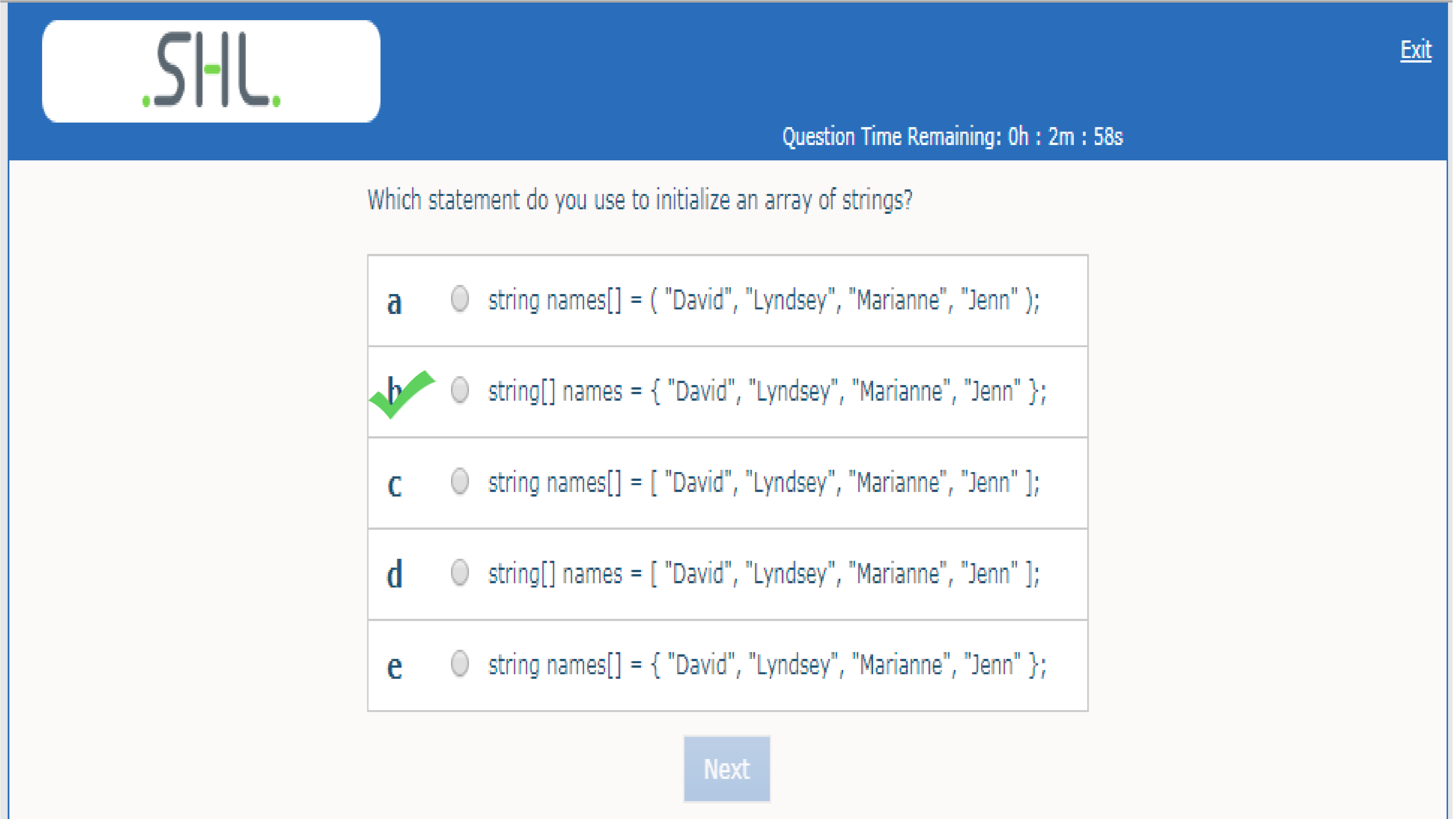
Question Time Remaining: 0h : 2m : 59s

```
string currentMethod = null;
Console.WriteLine((currentMethod ?? "not set").ToString());
```

What is written to the console when you execute the sample code above?

a ✓ ○	not set
b ○	nullnot set
c ○	currentMethod
d ○	currentMethod not set
e ○	null

Next



.SHL.

Exit

Question Time Remaining: 0h : 2m : 58s

Which statement do you use to initialize an array of strings?

a `string names[] = ( "David", "Lyndsey", "Marianne", "Jenn" );`

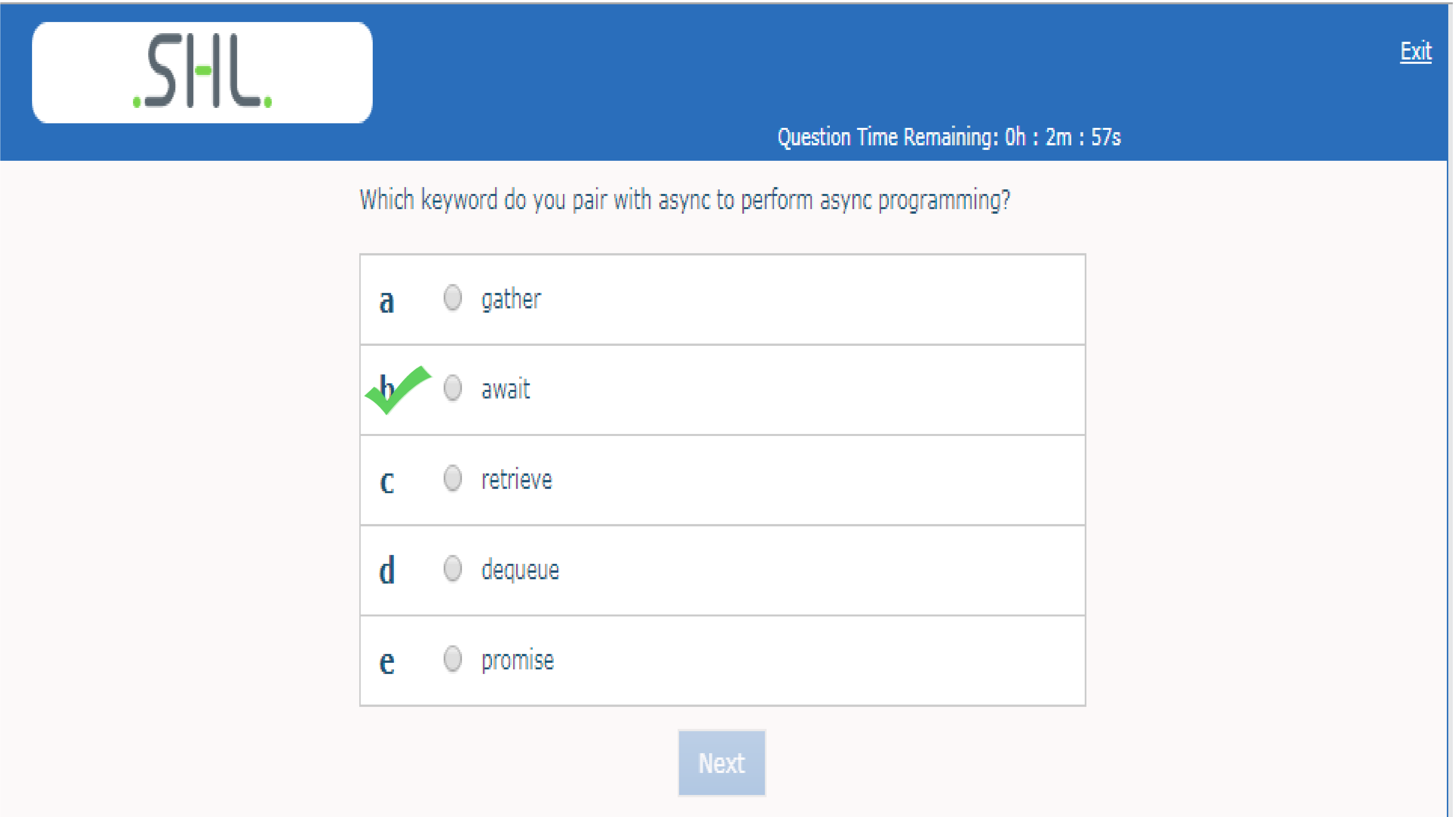
[✓ b `string[] names = { "David", "Lyndsey", "Marianne", "Jenn" };`]

c `string names[] = [ "David", "Lyndsey", "Marianne", "Jenn" ];`

d `string[] names = [ "David", "Lyndsey", "Marianne", "Jenn" ];`

e `string names[] = { "David", "Lyndsey", "Marianne", "Jenn" };`

Next



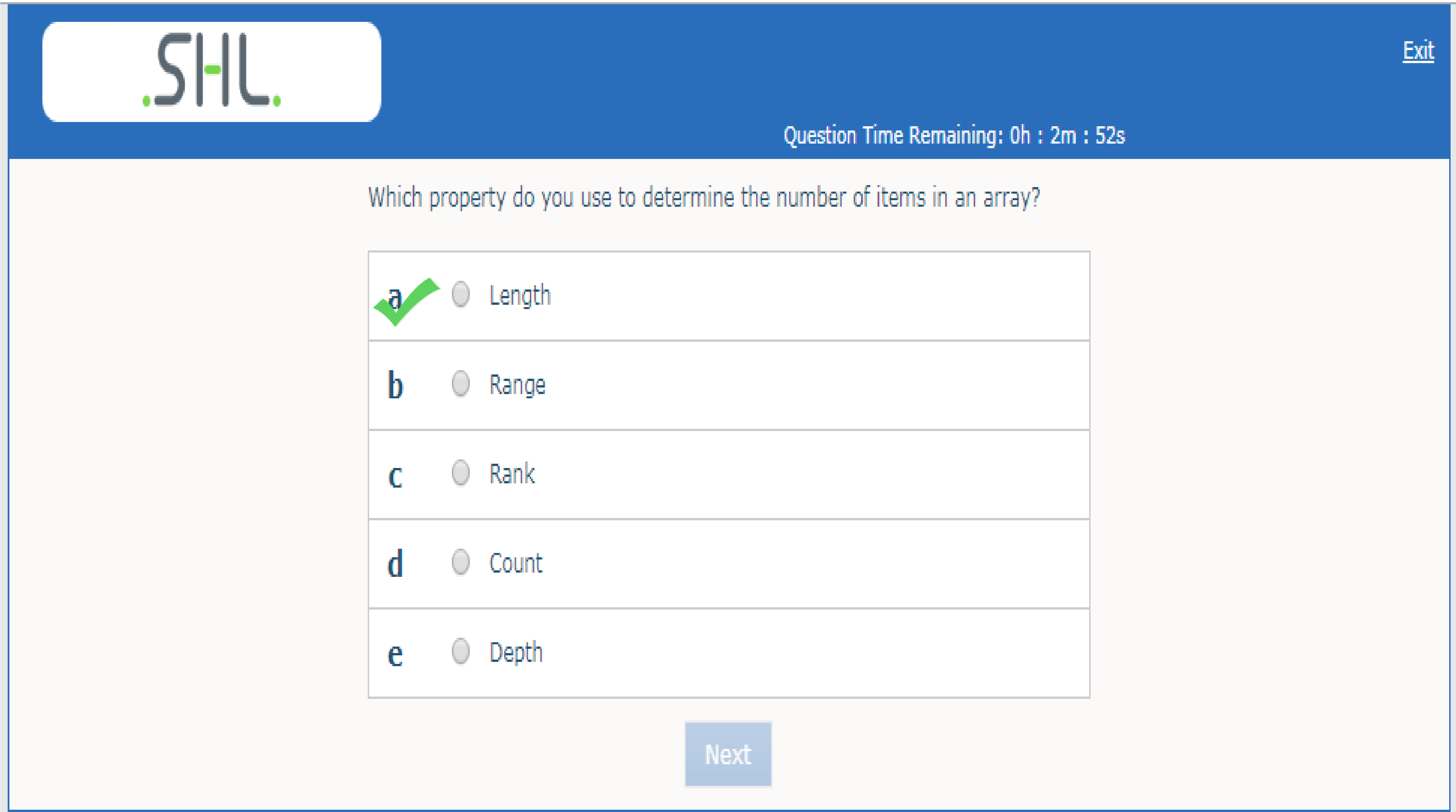
Exit

Question Time Remaining: 0h : 2m : 57s

Which keyword do you pair with async to perform async programming?

a	gather
[✓ b]	await
c	retrieve
d	dequeue
e	promise

Next


Exit

Question Time Remaining: 0h : 2m : 52s

Which property do you use to determine the number of items in an array?

a ✓	Length
b	Range
c	Rank
d	Count
e	Depth

Next