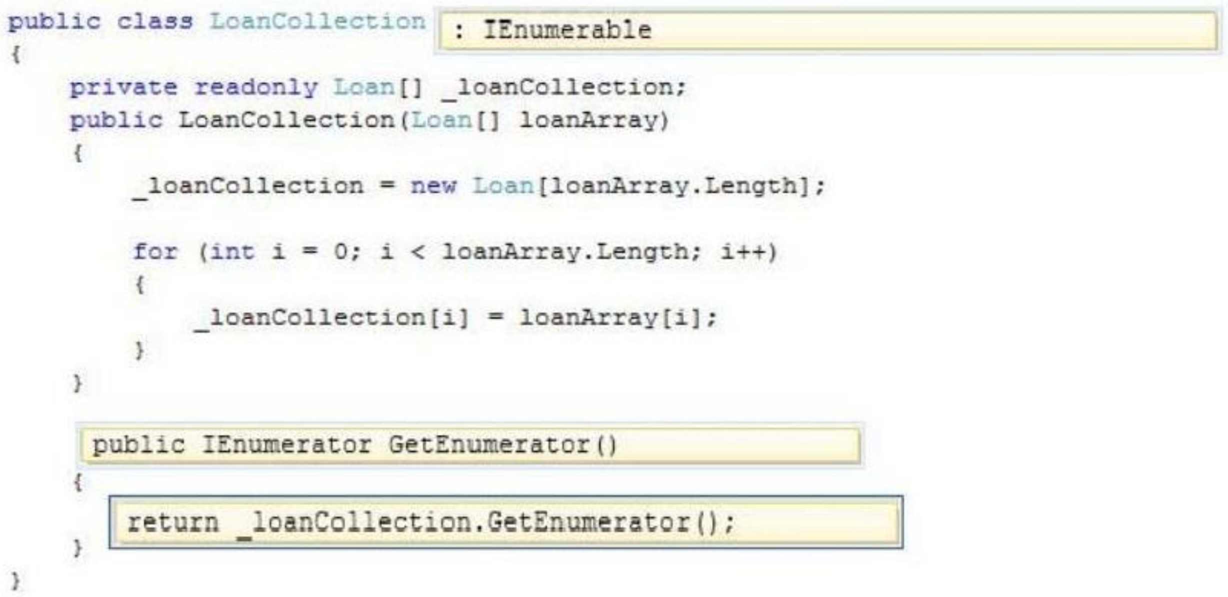
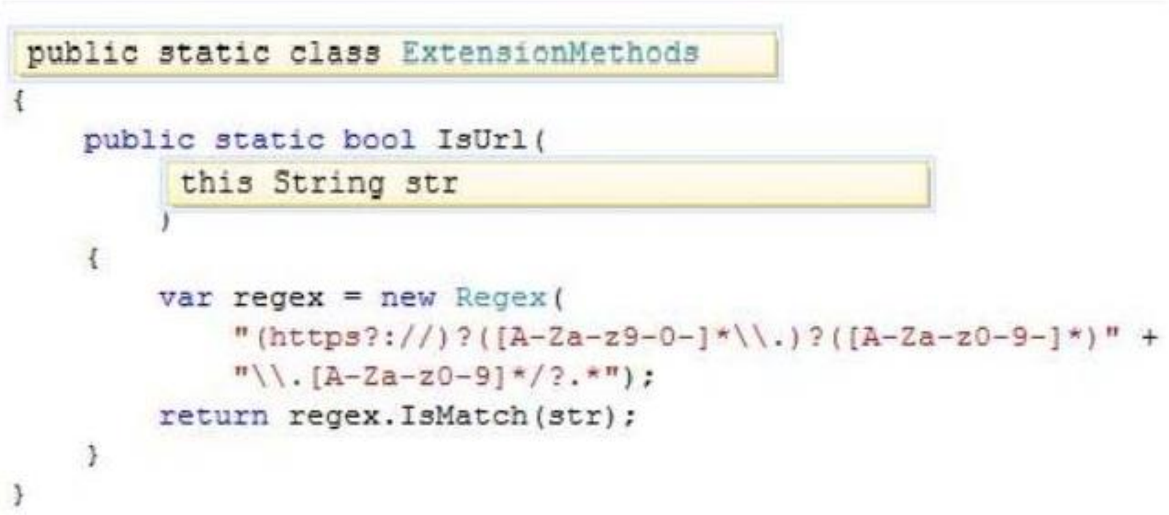
1. B



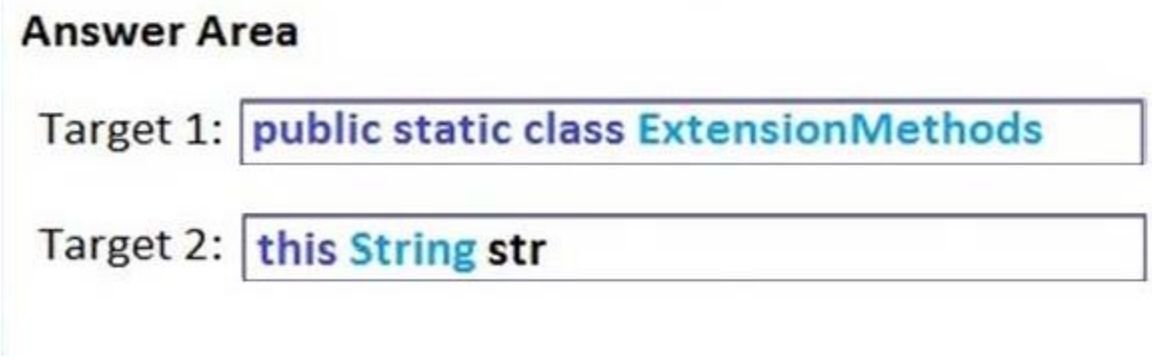
```
public class LoanCollection : IEnumerable
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }

    public IEnumerator GetEnumerator()
    {
        return _loanCollection.GetEnumerator();
    }
}
```

2.
3. D
4. C



```
public static class ExtensionMethods
{
    public static bool IsUrl(
        this String str
        )
    {
        var regex = new Regex(
            "(https?://)?([A-Za-z9-0-]*\\.)?([A-Za-z0-9-]*)" +
            "\\.[A-Za-z0-9]*/?.*");
        return regex.IsMatch(str);
    }
}
```

5.
6. BD



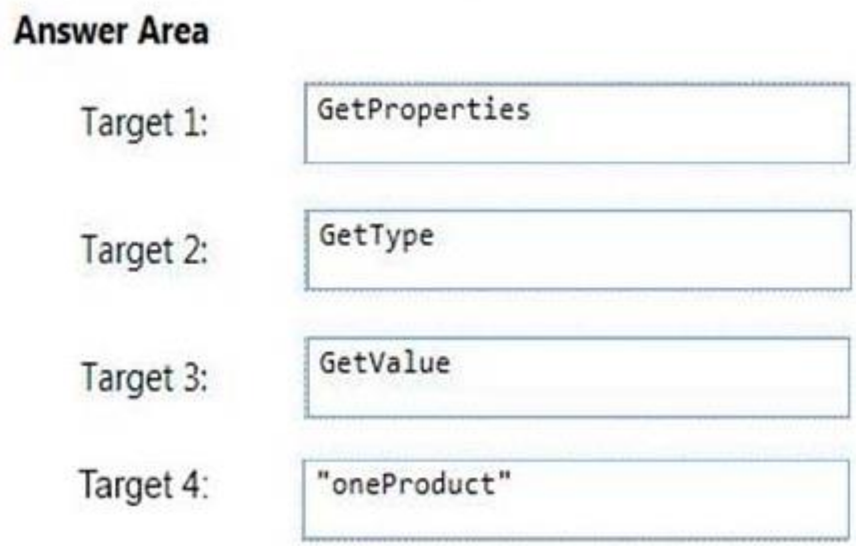
Answer Area

Target 1: public static class ExtensionMethods

Target 2: this String str

7.

8. A



Answer Area

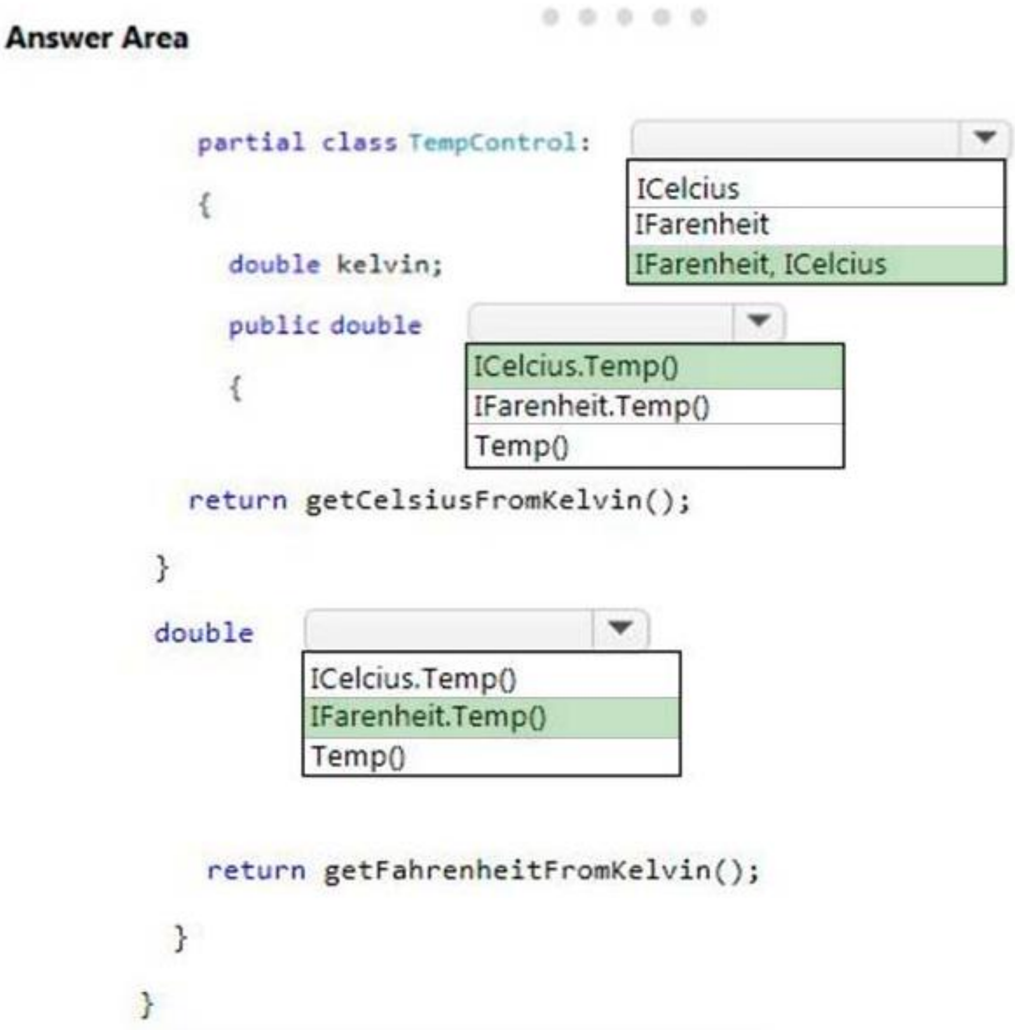
Target 1: GetProperties

Target 2: GetType

Target 3: GetValue

Target 4: "oneProduct"

9.
10. B



Answer Area

```
partial class TempControl:
```

ICelcius
IFarenheit
[✓ IFarenheit, ICelcius]

```
{
  double kelvin;
  public double
```

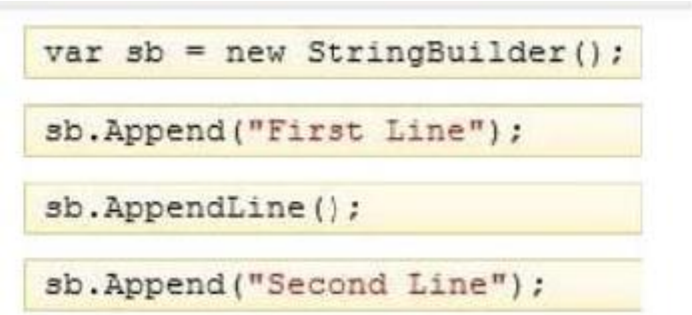
[✓ ICelcius.Temp()]
IFarenheit.Temp()
Temp()

```
  {
 return getCelsiusFromKelvin();
}
double
```

ICelcius.Temp()
[✓ IFarenheit.Temp()]
Temp()

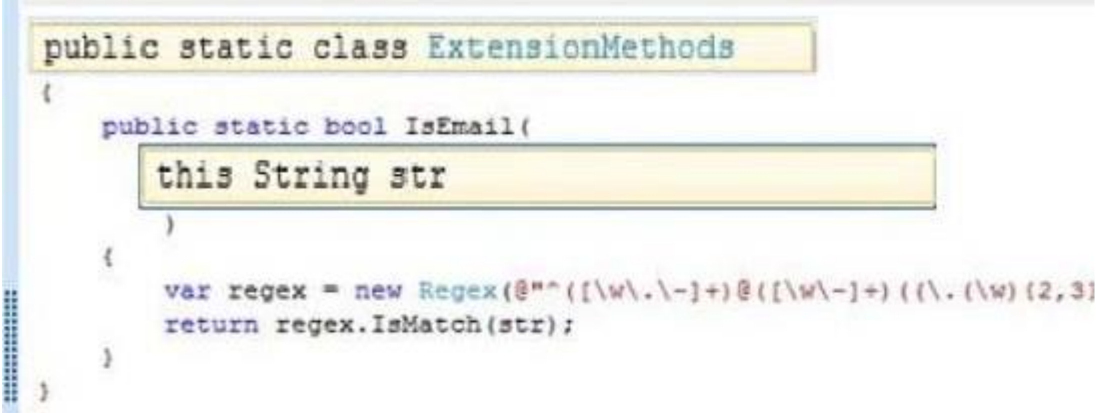
```
  return getFahrenheitFromKelvin();
 }
}
```

11.
12. C
13. D



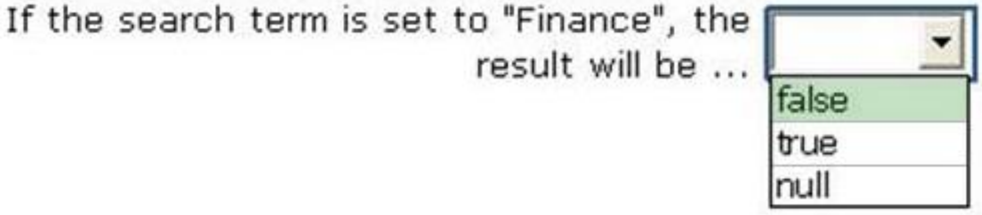
```
var sb = new StringBuilder();
sb.Append("First Line");
sb.AppendLine();
sb.Append("Second Line");
```

14.
15. AC



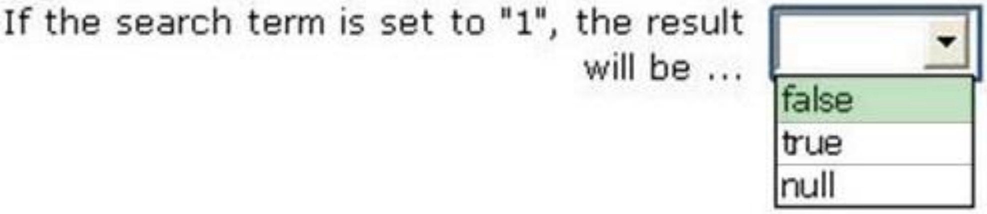
```
public static class ExtensionMethods
{
    public static bool IsEmail(
        this String str
        )
    {
        var regex = new Regex(@"^([\w\.\-]+)@([\w\-]+)((\.(\w){2,3}
        return regex.IsMatch(str);
    }
}
```

16.
17. E
18. A
19. B
20. D
21. D
22. EF
23. A



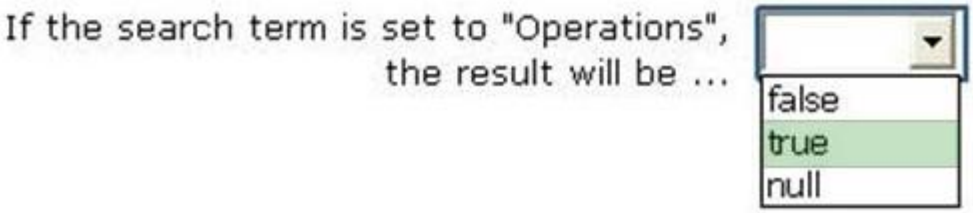
If the search term is set to "Finance", the result will be ...

[✓ false]
true
null



If the search term is set to "1", the result will be ...

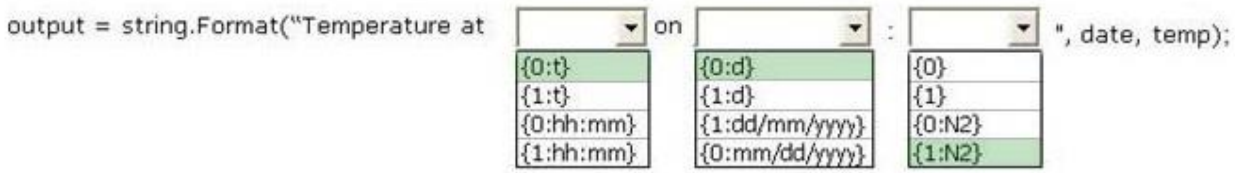
[✓ false]
true
null



If the search term is set to "Operations", the result will be ...

false
[✓ true]
null

24.
25. D



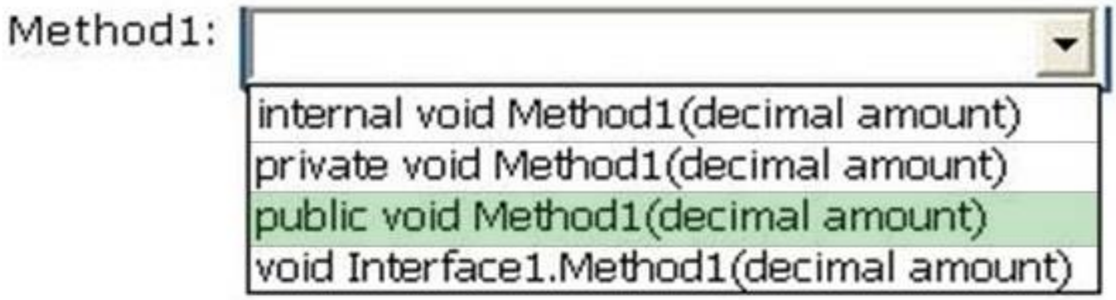
```
output = string.Format("Temperature at [{0:t}] on [{0:d}] : [{1:N2}] ", date, temp);
```

- [✓ {0:t}]
- {1:t}
- {0:hh:mm}
- {1:hh:mm}

- [✓ {0:d}]
- {1:d}
- {1:dd/mm/yyyy}
- {0:mm/dd/yyyy}

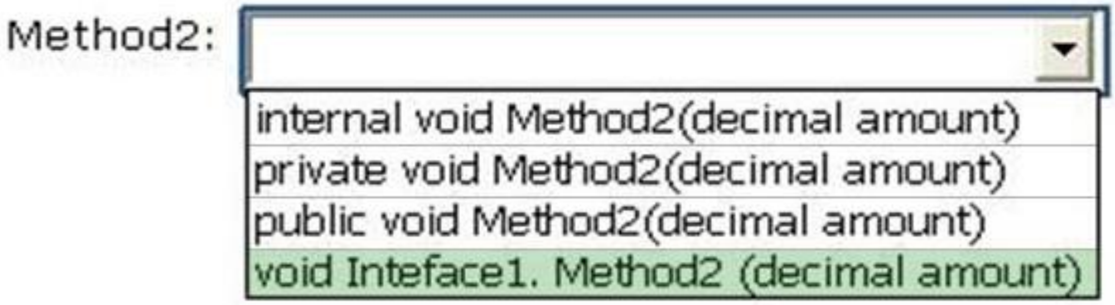
- {0}
- {1}
- {0:N2}
- [✓ {1:N2}]

26.
27. D
28. B



Method1:

- internal void Method1(decimal amount)
- private void Method1(decimal amount)
- [✓ public void Method1(decimal amount)]
- void Interface1.Method1(decimal amount)

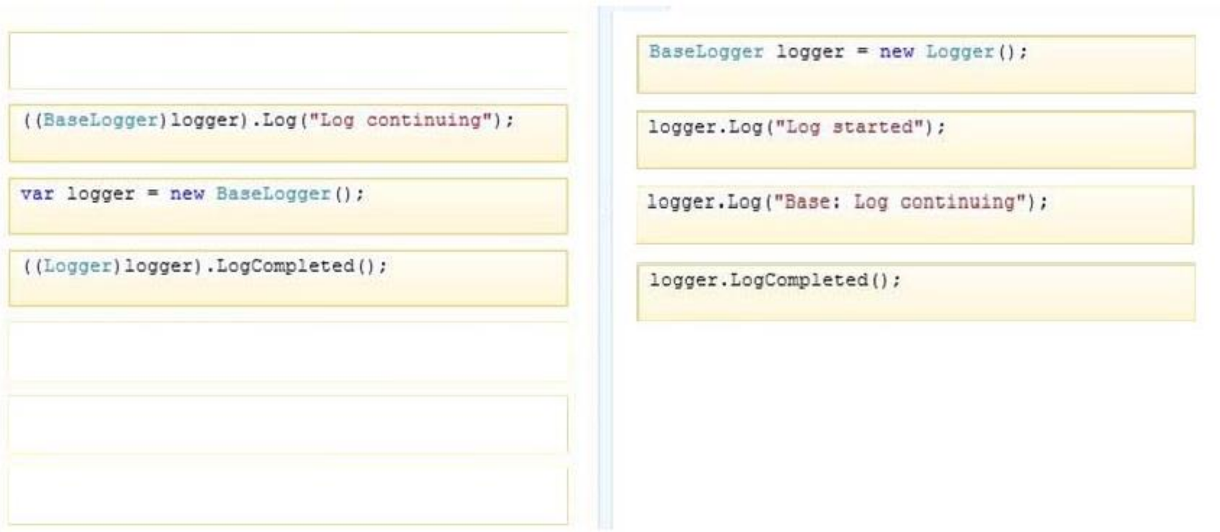


Method2:

- internal void Method2(decimal amount)
- private void Method2(decimal amount)
- public void Method2(decimal amount)
- [✓ void Inteface1. Method2 (decimal amount)]

29.
30. BC
31. A
32. A
33. C
34. A
35. C
36. D
37. B
38. D
39. C

40.



```
((BaseLogger)logger).Log("Log continuing");
```

```
var logger = new BaseLogger();
```

```
((Logger)logger).LogCompleted();
```

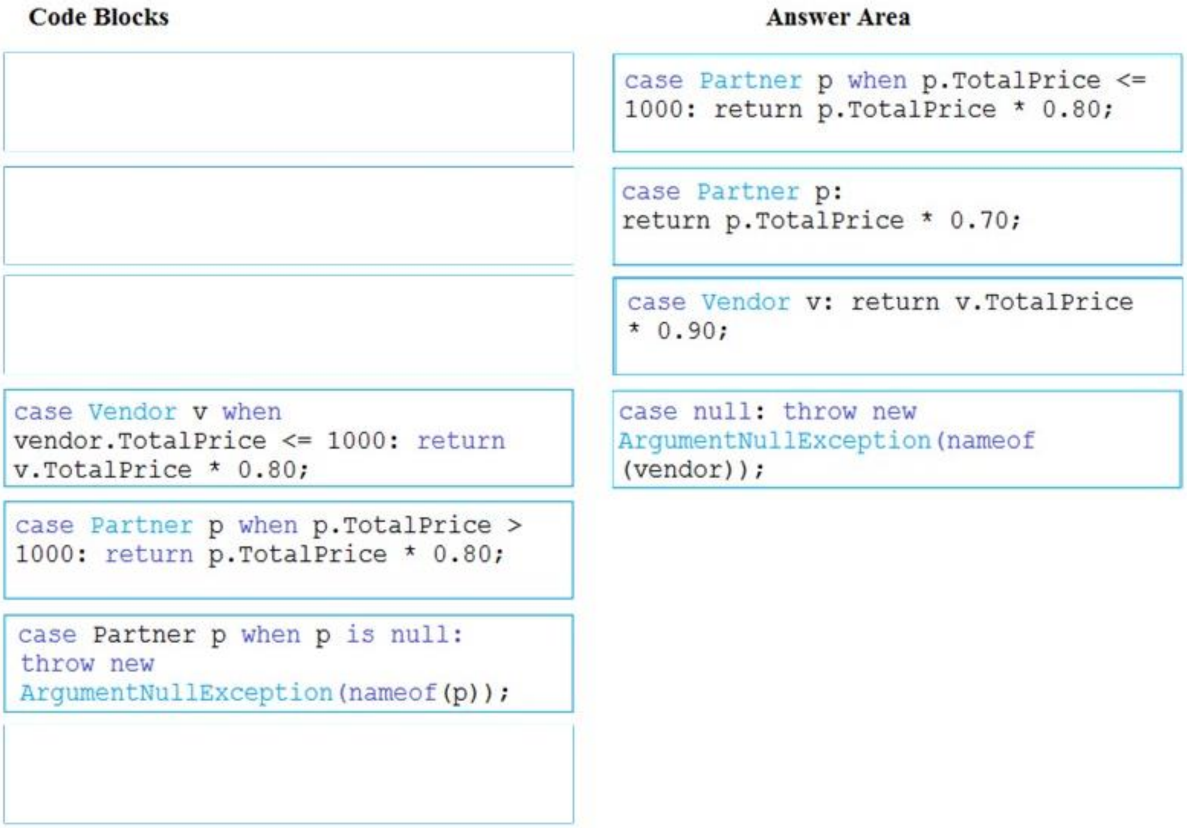
```
BaseLogger logger = new Logger();
```

```
logger.Log("Log started");
```

```
logger.Log("Base: Log continuing");
```

```
logger.LogCompleted();
```

41.



Code Blocks

```
case Vendor v when
vendor.TotalPrice <= 1000: return
v.TotalPrice * 0.80;
```

```
case Partner p when p.TotalPrice >
1000: return p.TotalPrice * 0.80;
```

```
case Partner p when p is null:
throw new
ArgumentNullException(nameof(p));
```

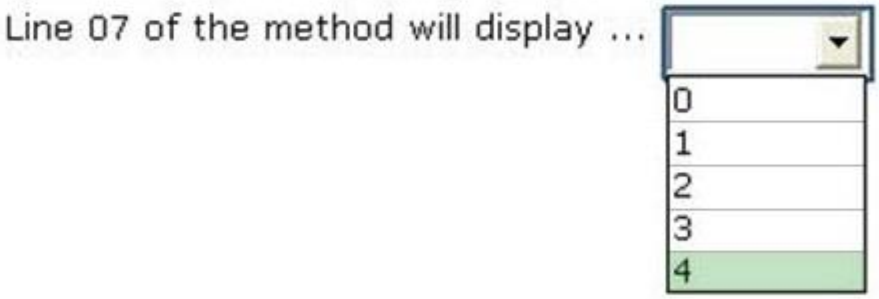
Answer Area

```
case Partner p when p.TotalPrice <=
1000: return p.TotalPrice * 0.80;
```

```
case Partner p:
return p.TotalPrice * 0.70;
```

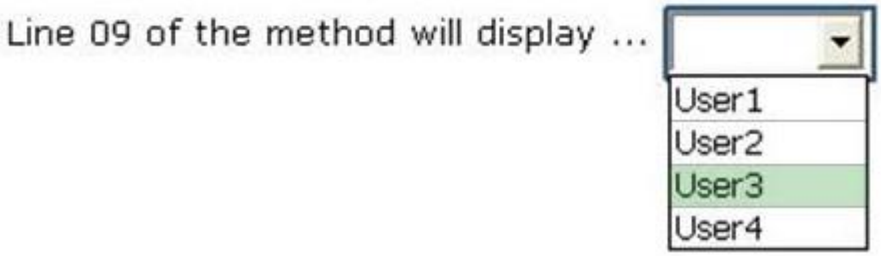
```
case Vendor v: return v.TotalPrice
* 0.90;
```

```
case null: throw new
ArgumentNullException(nameof
(vendor));
```



Line 07 of the method will display ...

0
1
2
3
[✓ 4]



Line 09 of the method will display ...

User1
User2
[✓ User3]
User4

42.
43. C
44. AC
45. B