



<u>Da</u>

Hide sidebars

# CIT195 LB [META 2302]

<u>Dashboard</u> > My courses > <u>Spring 2023 Courses</u> > <u>Business: Spring 2023 (2302)</u> > <u>CIT195 LB [META 2302]</u>

> Understanding the C# Object Model - Week 4 (3/20-4/3) > Unit 2 Quiz (Chapters 15 through 22)

		Course dashboard 🌼
Started on	Monday, 10 April 2023, 6:58 PM	
State	Finished	
Completed on	Monday, 10 April 2023, 8:12 PM	
Time taken	1 hour 13 mins	
Grade	<b>44.80</b> out of 50.00 ( <b>89.6</b> %)	

^

Hide sidebars

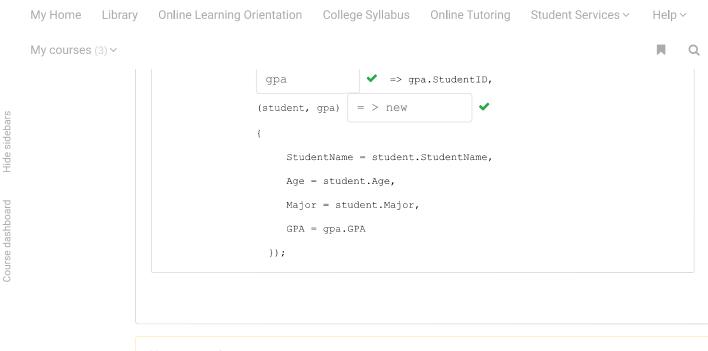
My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services ∨ Help ∨

My courses (3) ∨ 

Q

Configure the query to join the two lists below on StudentID and display the Student's Name, Age, Major, and GPA.

```
// Student collection
       IList < Student > studentList = new List < Student >() {
               new Student() { StudentID = 1, StudentName = "Frank Furter", Age = 55,
Major="Hospitality"} ,
                new Student() { StudentID = 2, StudentName = "Gina Host", Age = 41,
Major="Hospitality"} ,
               new Student() { StudentID = 3, StudentName = "Cookie Crumb", Age =
41, Major="CIT" } ,
               new Student() { StudentID = 4, StudentName = "Ima Script", Age = 18,
Major="CIT" } ,
               new Student() { StudentID = 5, StudentName = "Cora Coder", Age = 35,
Major="CIT" } ,
               new Student() { StudentID = 6, StudentName = "Ura Goodchild" , Age =
20, Major="Marketing"} ,
               new Student() { StudentID = 7, StudentName = "Take Mewith" , Age = 19,
Major="Aerospace Engineering" }
            };
        // Student GPA Collection
       IList < StudentGPA > studentGPAList = new List < StudentGPA > () {
               new StudentGPA() { StudentID = 1, GPA=4.0},
               new StudentGPA() { StudentID = 2, GPA=3.5} ,
               new StudentGPA() { StudentID = 3, GPA=2.0 } ,
               new StudentGPA() { StudentID = 4, GPA=1.5 } ,
               new StudentGPA() { StudentID = 5, GPA=4.0 } ,
               new StudentGPA() { StudentID = 6, GPA=2.5},
               new StudentGPA() { StudentID = 7, GPA=1.0 }
            };
```

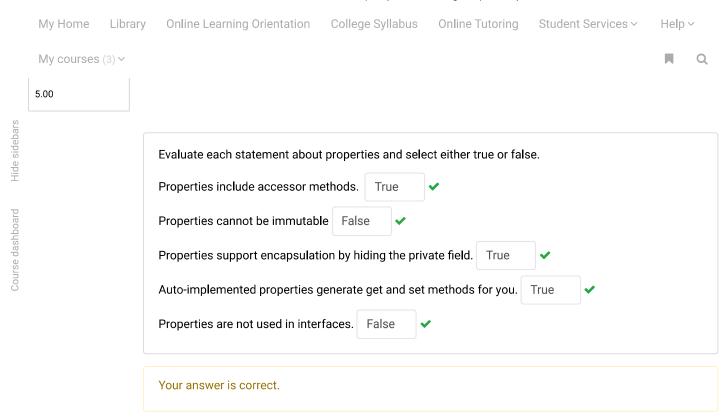


M Q

Help ~

d Hide sidebars

```
Evaluate each statement about delegates and select either true or false.
A delegate can be passed as a parameter to methods.
The maximum number of methods referenced by a single delegate is one.
                                                                        False
To invoke a delegate, use the delegate object and pass any parameters needed by the method it
references. True
In the code shown below, line #1 is declaring the delegate. False
01 Calculator c = new Calculator(Program.Addition);
02 Console.WriteLine($"Addition of 5 and 10 is : {c(5,10)}");
In the code shown below, a Calc delegate can be used for either method when it is instantiated.
 False
     class Numbers
          public delegate void Calc(int x);
          public static int num { get; set; } = 0;
          public static void Addition(int a)
               num += a;
          public static int Multiplication(int a, int b)
              int num = a * b;
              return num;
```



https://elearn.nmc.edu/mod/quiz/review.php?attempt=1784782&cmid=2134306



Evaluate each statement about records and select either true or false.

The record declaration below allows you to directly access and change any value using the set

accessor. False

public record Person(string FirstName, string LastName);

You can print the record object to the console using the object name to see the properties and their values. True

Using a record object, you can directly access and display any value using the get accessor.



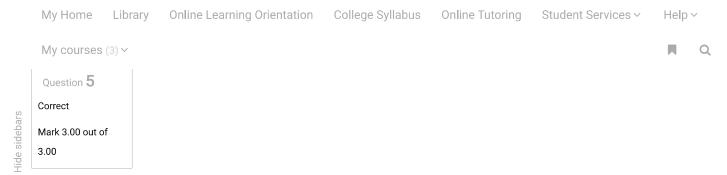
#### Example:

```
public record Person(string FirstName, string LastName);

public static void Main()
{
    Person person = new("Scooby", "Doo");
    Console.WriteLine($"First name={person.FirstName}");
    Console.WriteLine($"Last name={person.LastName}");
}
```

The record declaration below is non-positional. True

```
public record Person()
{
    public string FirstName{ get; init; }
    public string LastName { get; init; }
    public string City { get; init; }
    public string State { get; init; }
    public string PostalCode { get; init; }
}
```



Course dashboard

Q

Help ~

5.00

My courses (3) ~

Select the correct type of collection based on the description given.		
A collection of values that can be identified and retrieved by using keys rather than indexes.	Dictionary	
A first-in, last-out (FILO) data structure with methods to add an item onto the top of the structure, remove an item from the top of the structure, and examine the item at the top of the structure without removing it.	Stack	
A double-ended ordered list, optimized to support insertion and removal at either end. This collection can be accessed FIFO or FILO, and it supports random access.	Linked List	
A first-in, first-out data structure, with methods to add an item to one end of the structure, remove an item from the other end, and examine an item without removing it.	Queue	
An unordered set of values that is optimized for fast retrieval of data. It provides set-oriented methods for comparing, combining, or intersecting data sets.	Hash Set	

Hide sidebars

Given the class shown below, how would you instantiate a class object for an integer array with a size of 10?

```
public class MyGenericArray < T >
{
    private T[] array;

    public MyGenericArray(int size)
    {
        array = new T[size + 1];
    }
    public T getItem(int index)
    {
        return array[index];
    }
    public void setItem(int index, T value)
    {
        array[index] = value;
    }
}
```

```
int[] intArray = new int[10];
```

```
C.
    MyGenericArray < int > intArray = new MyGenericArray;
```

```
MyGenericArray < int > intArray = new MyGenericArray < int >(10);
```

Your answer is correct.

Q

١.

### Given the code shown below, which line declares the delegate?

```
namespace test2
  public delegate void Notify();
   public class myBusiness
       public event Notify Done;
       public void StartProcess()
           Console.WriteLine("A long day of meetings has begun.");
           OnProcessCompleted();
       protected virtual void OnProcessCompleted()
           Done?.Invoke();
   class Program
       public static void Main()
           myBusiness meeting = new myBusiness();
           meeting.Done += EndOfDay;
           meeting.StartProcess();
       public static void EndOfDay()
           Console.WriteLine("Done for the day!");
```

- a. public event Notify Done;
- b. public delegate void Notify();
- o. protected virtual void OnProcessCompleted() { Done?.Invoke(); }
- d. meeting.Done += EndOfDay;

Q

My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services > Help >

My courses (3) >

Hide sidebars

# Given the code shown below, which line registers the event?

```
namespace test2
  public delegate void Notify();
   public class myBusiness
       public event Notify Done;
       public void StartProcess()
           Console.WriteLine("A long day of meetings has begun.");
           OnProcessCompleted();
       protected virtual void OnProcessCompleted()
           Done?.Invoke();
   class Program
       public static void Main()
           myBusiness meeting = new myBusiness();
           meeting.Done += EndOfDay;
           meeting.StartProcess();
       public static void EndOfDay()
           Console.WriteLine("Done for the day!");
```

- a. protected virtual void OnProcessCompleted() { Done?.Invoke(); }
- b. meeting.Done += EndOfDay;
- c. public delegate void Notify();
- d. public event Notify Done;

Your answer is incorrect.

Q

My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services ∨ Help ∨

My courses (3) ∨ ■ Q

Hide sidebars

## Given the studentList shown below, which query counts students who are over 18 years old?

```
IList < Student > studentList = new List < Student >() {
     new Student() { StudentID = 1, StudentName = "Frank Furter", Age = 55,
Major="Hospitality", Tuition=3500.00},
     new Student() { StudentID = 1, StudentName = "Gina Host", Age = 21,
Major="Hospitality", Tuition=4500.00 } ,
      new Student() { StudentID = 2, StudentName = "Cookie Crumb", Age = 21,
Major="CIT", Tuition=2500.00 } ,
     new Student() { StudentID = 3, StudentName = "Ima Script", Age = 48,
Major="CIT", Tuition=5500.00 } ,
     new Student() { StudentID = 3, StudentName = "Cora Coder", Age = 35,
Major="CIT", Tuition=1500.00 } ,
     new Student() { StudentID = 4, StudentName = "Ura Goodchild" , Age = 40,
Major="Marketing", Tuition=500.00},
     new Student() { StudentID = 5, StudentName = "Take Mewith", Age = 29,
Major="Aerospace Engineering", Tuition=5500.00 }
};
```

- a. countAge = from s in studentList where s.Age>18 s.Count();
- b. countAge = studentList.Count();
- C. countAge = studentList.Count(s=>s.Age>18);
- d. countAge = studentList.Where(s=>s.Age>18).Count(s => s.Age);

Your answer is incorrect.

Q

My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services ∨ Help ∨

My courses (3) ∨ ■ Q

Hide sidebars

```
Select the statement that creates the indexer in the following example
```

```
class Gym
{
    private string[] members = new string[100];
    public string type {get;set;}
    public double price {get,set;}
    public string this[int i]
    {
        get
        {
            return names[i];
        }
        set
        {
            names[i] = value;
        }
    }
}
```

```
public string this[int i]
{
    get
    {
        return names[i];
    }
    set
    {
        names[i] = value;
    }
}
```

```
public double price {get;set;}
```

```
O C. public string type {get;set;}
```

	My Home	Library	Online Learning Orientation	College Syllabus	Online Tutoring	Student Services >	Help	Help ~	
	My courses	(3) ~						Q	
d Hide sidebars			O e. class Gym						
Course dashboard	Question 12 Correct								
	Mark 1.00 out o 1.00	f							

```
What type of builtin delegate should you use for the method below?

static int Sum(int x, int y)
{
    return x + y;
}

a. Func

b. Action

c. Event delegate

d. Custom delegate
```

Help ~

Q

2.00

Hide sidebars

```
Which 2 queries will retrieve C# Tutorials from the list below? Choose 2.
 IList < string > stringList = new List < string > ()
 "C# Tutorials",
 "Advanced C# Tutorials",
 "LINQ Query Tutorials",
 "Learn C++",
 "MVC Tutorial" ,
 "MVC C# Tutorials",
 "Beginning RazorPages"
 };
__ a.
        var result = stringList.Where(s => "C# Tutorials");
□ b.
        var result = from s in stringList
                      where s.Contains("C# Tutorials")
                      select s;
var result = from s in stringList
                      .Where(s => s.Contains("C# Tutorials"))
                       select s;
 ✓ d.
        var result = stringList.Where(s => s.Contains("C# Tutorials"));
```

Your answer is partially correct.

You have correctly selected 1.

Help ~

Q

My courses (3) ~

1.00

Hide sidebars

Course dashboard

## Which query groups the list below by Age?

```
IList studentList = new List() {
       new Student() { StudentID = 1, StudentName = "Frank Furter", Age = 55,
Major="Hospitality"} ,
       new Student() { StudentID = 1, StudentName = "Gina Host", Age = 41,
Major="Hospitality"} ,
        new Student() { StudentID = 2, StudentName = "Cookie Crumb", Age = 21,
Major="CIT" } ,
       new Student() { StudentID = 3, StudentName = "Ima Script", Age = 38,
Major="CIT" } ,
       new Student() { StudentID = 3, StudentName = "Cora Coder", Age = 35,
Major="CIT" } ,
       new Student() { StudentID = 4, StudentName = "Ura Goodchild" , Age = 20,
Major="Marketing"} ,
       new Student() { StudentID = 5, StudentName = "Take Mewith", Age = 19,
Major="Aerospace Engineering" }
};
```

- a. var groupedAge = studentList.OrderBy(o=>o.Age).GroupBy(s => s.Major);
- b. var groupedAge = from s in studentList orderby s.StudentName group s by s.Major;
- C. var groupedAge = from s in studentList orderby s.Age group s by s.ID;
- d. var groupedAge = from s in studentList orderby s.Age group s by s.Major;

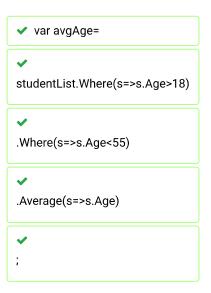
My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services ∨ Help ∨

My courses (3) ∨

2.00

Move the code to create a query that will retrieve the average age of students over 18 and below 55 for the list shown below:

```
IList < Student > studentList = new List < Student >() {
     new Student() { StudentID = 1, StudentName = "Frank Furter", Age = 17,
Major="Hospitality", Tuition=3500.00},
     new Student() { StudentID = 1, StudentName = "Gina Host", Age = 21,
Major="Hospitality", Tuition=4500.00 } ,
     new Student() { StudentID = 2, StudentName = "Cookie Crumb", Age = 21,
Major="CIT", Tuition=2500.00 } ,
     new Student() { StudentID = 3, StudentName = "Ima Script", Age = 18,
Major="CIT", Tuition=5500.00 } ,
      new Student() { StudentID = 3, StudentName = "Cora Coder", Age = 65,
Major="CIT", Tuition=1500.00 } ,
     new Student() { StudentID = 4, StudentName = "Ura Goodchild" , Age = 40,
Major="Marketing", Tuition=500.00},
     new Student() { StudentID = 5, StudentName = "Take Mewith" , Age = 29,
Major="Aerospace Engineering", Tuition=5500.00 }
};
```



My Home Library Online Learning Orientation College Syllabus Online Tutoring Student Services ∨ Help ∨

My courses (3) ∨

1.00

Given the code segment shown below, which line invokes the operator == method?

```
01
      Calculator first = new Calculator();
02
      Calculator second= new Calculator();
03
      first.number = r.Next(10, 20);
04
      second.number = r.Next(10, 20);
05
      Console.WriteLine($"Number1 = {first.number} and Number2= {second.number}");
      Console.WriteLine($"Is {first.number} = {second.number}? {first == second}");
06
07
     Console.WriteLine($"Is {first.number} != {second.number}? {first != second}");
      Console.WriteLine($"Is {first.number} > {second.number}? {first > second}");
08
      {\tt Console.WriteLine(\$"Is \{first.number\} < \{second.number\}? } \\ {\tt first < second}");
09
```

- a. 09
- o b. 05
- oc. 08
- d. 06
   ✓
- e. 07

	My Home	Library	Online Learning Orientation	College Syllabus	Online Tutoring	Student Services >	Help	~
	My courses	(3) ~						Q
	2.00							
Hide sidebars			Which 2 operator methods mu	st be overloaded tog	ether? Choose 2.			
			a. operator ++					
Course dashboard			■ b. operator +					
e dash			c. operator -					
Cours			☑ d. operator > ✓					
			☑ e. operator <❤					
			f. operator					
			Your answer is correct.					
			Tour answer is correct.					

Question 18

Incorrect

Mark 0.00 out of

1.00

```
Which operator is overloaded in the code segment shown below?

Trip day1 = new Trip(new DateTime(2023,10,1),100,8.5f);

Trip day2 = new Trip(new DateTime(2023,10,2), 100, 7.5f);

Trip TotalTrip = new Trip();

TotalTrip += day1;

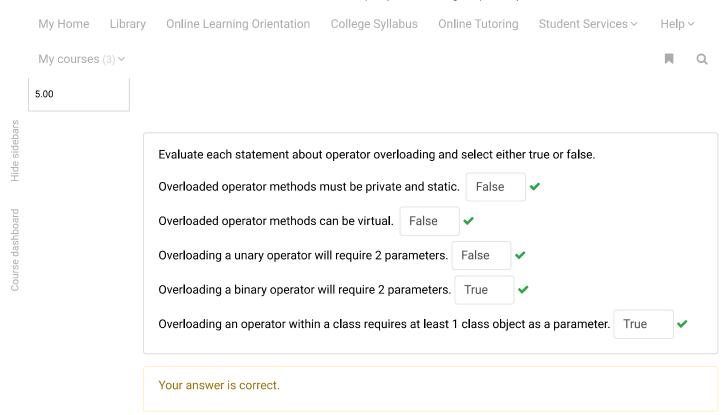
TotalTrip+= day2;

a. operator = *

b. operator +=

c. operator ++

d. operator +
```



Hide sidebars

You are logged in as Mike Tonder (Log out) Get the mobile app