**Topic 2: Constructors and Overloading**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286781_1)

Object-Oriented Programming C#

* + Microsoft:docs.microsoft.com
    - [Constructors (C# Programming guide)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/object-oriented-programming)
    - [Using Constructors (C# Programming guide)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/using-constructors)
  + GeeksForGeeks: geeksforgeeks.com
    - [C# | Constructors](https://www.geeksforgeeks.org/c-sharp-constructors/)
  + TutorialsPoint: tutorialspoint.com
    - [Constructors in C#](https://www.tutorialspoint.com/constructors-in-chash)
  + W3Schools: w3schools.com
    - [C# Constructors](https://www.w3schools.com/cs/cs_constructors.asp)

**Constructors**

[Constructors](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286781_1)

**Constructor, Constructor, what's your function?**  
Making some objects with properties and methods.  
**Constructor, Constructor, what's your function?**   
I got three favorite cars  
I call my constructor thrice not once.  
**Constructor, Constructor, what's that function?**   
I got "new" memory allocation.   
They'll get you one car.

(For those who do not know American Saturday morning cartoons that teach you things, fortunately the internet can inform us)

[Watch Video](http://www.youtube.com/watch?v=RPoBE-E8VOc)

**Schoolhouse Rock: Grammar - Conjunction Junction Music Video**

**Duration:** 3:23   
**User:** n/a - **Added:** 12/8/11

**But seriously... Constructors**

[But seriously... Constructors](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286781_1)

Constructors make a new instance (an object) of a class. So recall the House class Default No-Arg Constructor

// Default No-Arg Constructor

public House()

{

Address = "";

NumberBedrooms = 0;

NnumberBathrooms = 0;

}

Using the Default No-Arg Constructor, in a method (such as Main) the following code will create a new object of type House and set the properties.

// Create a House Object

House myHouse = new House(); // notice the keyword new and the call to the constructor House()

// Set the object's Address property

myHouse.Address = "123 Main Street";

// Set the object's NumberBathrooms property

myHouse.NumberBedrooms = 3;

// Set the object's NumberBedrooms property

myHouse NumberBathrooms = 2;

**Overloading**

[Overloading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286781_1)

Overloading in English means to give too much demand, but in programming it is not too great demand to overload, it's part of most programs.

Overloading means you have different behaviors based on definition and use. You have already written and used overloaded methods and used overloaded operators.

Plus (**+**) was both integer additions and string concatenation.

string FirstName = "Tatum";

string LastName = "Channing";

string Name = FirstName + " " + LastName;

int x = 4 + 5;

decimal y = 5.0 + 6.0;

[Check out more on C# + operator.](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/addition-operator)

Methods can also be overloaded, such as the same method having different types and/or number of parameters.

public int add(int x, int y) // two parameters, both int

{

return x + y;

}

public int add(int x, int y,int z) //three parameters, all int

{

return x + y + z;

}

public float add(float x, float y, float z) //three parameters, all float

{

return x + y + z;

}

**Overloading Constructors**

[Overloading Constructors](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286781_1)

Even constructors can be overloaded and often are. Thus you can make an object and set its properties at the same time.

// Create a House Object

House myHouse = new House();

// Set the object's Address property

myHouse.Address = "123 Main Street";

// Set the object's NumberBathrooms property

myHouse.NumberBedrooms = 3;

// Set the object's NumberBedrooms property

myHouse NumberBathrooms = 2;

Can become the following call:

// Create a House Object and set properties

House myHouse = new House("123 Main Street", 3, 2);

When you add the overloaded Constructor below to the House class:

public House(string houseAddress, int numBedrooms, int numBathrooms)

{

Address = houseAddress;

NumberBedrooms = numBedrooms;

NumberBathrooms = numBathrooms;

}

[**Overloading Constructors Quiz**](https://dmacc.blackboard.com/webapps/blackboard/content/launchAssessment.jsp?course_id=_102593_1&content_id=_7286804_1&mode=view)

This quiz covers overloading constructor for a class. It is not timed, you have 2 attempts.