

Instantiating Classes

Instantiating Classes

A class is just a blueprint for a type. To use the behaviors and characteristics that you define within a class, you need to create instances of the class. An instance of a class is called an object.

To create a new instance of a class, you use the `new` keyword, as shown in the following example:

```
// Instantiating a Class  
DrinksMachine dm = new DrinksMachine();
```

When you instantiate a class in this way, you are actually doing two things:

- You are creating a new object in memory based on the `DrinksMachine` type.
- You are creating an object reference named `dm` that refers to the new `DrinksMachine` object.

When you create your object reference, instead of explicitly specifying the `DrinksMachine` type, you can allow the compiler to deduce the type of the object at compile time. This is known as type inference. To use type inference, you create your object reference by using the `var` keyword, as shown in the following example:

```
// Instantiating a Class by Using Type Inference  
var dm = new DrinksMachine();
```

In this case, the compiler does not know in advance the type of the `dm` variable. When the `dm` variable is initialized as a reference to a `DrinksMachine` object, the compiler deduces that the type of `dm` is `DrinksMachine`. Using type inference in this way causes no change in how your application runs, it is simply a shortcut for you to avoid typing the class name twice. In some circumstances, type inference can make your code easier to read, while in other circumstances it may make your code more confusing. As a general rule, consider using type inference when the type of variable is absolutely clear.

After you have instantiated your object, you can use any of the members—methods, fields, properties, and events—that you defined within the class, as shown in the following example:

```
// Using Object Members  
var dm = new DrinksMachine();  
dm.Make = "Fourth Coffee";  
dm.Model = "Beancrusher 3000";  
dm.Age = 2;  
dm.MakeEspresso();
```

This approach to calling members on an instance variable is known as dot notation. You type the variable name, followed by a period, followed by the member name. The IntelliSense feature in Visual Studio will prompt you with member names when you type a period after a variable.

[Learn About Verified Certificates](#)

© All Rights Reserved