**Polymorphism** allows [methods](https://www.codecademy.com/resources/docs/c-sharp/methods) with the same name to perform different actions depending on the object.  
- Consider an Animal class with subclasses like Dog, Cat, and Bird. Polymorphism lets you call the same method, such as MakeSound(), on any of these objects and get different results based on the subclass.

**Polymorphism Benefits:**

1. **Flexibility and Maintainability:** You can add new [classes](https://www.codecademy.com/resources/docs/c-sharp/classes) with little or no modification to the existing code.
2. **Code Reusability:** Common functionality can be written in the base class, and specific functionality can be implemented in derived classes.
3. **Simplified Code:** Complex switch-case or if-else structures can be avoided, making the code cleaner and easier to read.

**Virtual and Overriding Methods:**

- A virtual method is a method in the base class that can be overridden in derived [classes](https://www.codecademy.com/resources/docs/c-sharp/classes). The virtual keyword is used to allow derived classes to provide specific implementations of this method.

A computer screen shot of text

Description automatically generated

- Method overriding allows a derived class to provide a specific implementation of a method that is already defined in its base class. The override keyword is used to indicate that a method in a derived class is intended to replace a method in the base class.

A computer code with text

Description automatically generated

**Upcasting Objects:**

- Used when we want to create an instance of our base class with access to the overridden [methods](https://www.codecademy.com/resources/docs/c-sharp/methods) of a derived class.   
- **Upcasting** in C# refers to converting a derived class instance to a base class instance. To upcast, we assign a derived instance to a variable with the base class type.  
- The upcasting is implicit, which means it happens automatically with no extra syntax.  
- An instance that has been upcast to a base class instance will have access to the following:

A blue background with white text

Description automatically generated A black background with white text

Description automatically generated

**Downcasting Objects:**

- Refers to converting an upcast instance to a derived one.   
- This process allows us to access the derived class members that are unavailable in the base class.  
- Downcasting is an explicit operation and requires extra syntax:

A screen shot of a computer

Description automatically generated

**Using ‘Is’ Operators:**

- The is operator checks if an object is compatible with a given type  
- The result of the is operator is true if an object can be upcast or downcast to a specified type and false if it can not.

A screen shot of a computer code

Description automatically generated

**Using ‘As’ Operators:**

- The as operator attempts to cast an object to a given type, returning null if the cast fails.   
- This is useful for safely trying to downcast an object to a more derived type without risking an exception.  
- It is important to note that the as operator can be used to upcast as well, but it is more common to use the implicit assignment statement when upcasting.

A computer screen with text

Description automatically generated

**Abstract Classes:**

- An **abstract class** provides a blueprint of what derived [classes](https://www.codecademy.com/resources/docs/c-sharp/classes) need to implement.   
- This is done through the use of implemented and unimplemented [methods](https://www.codecademy.com/resources/docs/c-sharp/methods).  
- Abstract classes can not be instantiated, but like all base classes, they can supply implemented methods. These methods can be virtual for derived classes to override.  
- Abstract classes can also contain **abstract methods**, which are not implemented in the abstract class and therefore must be implemented by the derived class.  
- The abstract keyword is used to specify both abstract classes and methods.

A screen shot of a computer program

Description automatically generated