

McKenna Johnson

CSE 210

04/05/2025

Inheritance Explanation

Inheritance in C# is an additional option to use with classes and methods. It essentially means that if you have one “parent” class (think something like ‘book’), and several “child” classes (like ‘children’s book’, ‘reference book’, etc....) there is a high likelihood that there will be overlapping methods (like writing out the name and author). In order to simplify and reduce redundancy in your code (which is a big benefit of inheritance), inheritance means that you can write the universal methods, stick them in the ‘parent’ class, and then simply access them in the ‘child’ classes when necessary. This can be done by using things like `:base()`, `:ParentClass`, or using the term *protected* instead of *private* for variables. This would be quite helpful when creating programs that have a similar core structure, but varying class behaviors. An example of inheritance can be seen below:

```
public class Breathing : Activity //Connected to the “Activity” parent class
```

```
{
```

```
    public Breathing()
```

```
{
```

```
    SetActivityName("Breathing"); //Method accessed from the “Activity” parent class
```

```
    SetActivityDescription("This program will walk you through a series of timed breaths. Inhale and exhale when prompted. Focus on relaxation and steadiness.");
```

```
}
```