From what I understand, it’s similar to building with Legos in that you can create new objects out of the existing pieces. For this to apply in programming, one class may inherit the attributes and functions of another class thanks to inheritance, this is key concept in object-oriented programming. The class being inherited from is referred to as the base class or parent class, while the class that inherits is known as the derived class or child class.

One of the main benefits of inheritance is code reusability, which saves time and effort when writing code. With inheritance, a new class can inherit the properties and methods of an existing class, eliminating the need to rewrite code that already exists in the parent class.

The class "Activity" is the parent class in the mindfulness program created this week, and the child classes "BreathingActivity," "ReflectionActivity," "ListingActivity," and "ReminderActivity" all derive from it. Code reuse is made feasible and time and effort are saved because the child classes have access to the parent class's properties and functions.

All the activities in the program have some things in common, like a name and description. Instead of writing the same code for each activity, I made an "Activity" template that has these basic things. Then, created new activities that "inherit" these things from the template and added new things specific to each activity, like a breathing exercise for the BreathingActivity.

Here's an example of inheritance from the program Mindfulness:

class Activity

{

// ...

public virtual void Start()

{

// ...

}

// ...

}

class BreathingActivity : Activity

{

// ...

}

class ReflectionActivity : Activity

{

// ...

}

class ListingActivity : Activity

{

// ...

}

class ReminderActivity : Activity

{

// ...

}

In this example, the child classes `BreathingActivity`, `ReflectionActivity`, `ListingActivity`, and `ReminderActivity` all inherit from the parent class `Activity`, which means they have access to the properties and methods defined in the parent class. For example, all of the child classes have a `Start` method, which is defined in the parent class.