Inheritance in a program is used to create base and subclasses. The base class contains information that is relevant between all different subclasses of the base. Meanwhile, the subclass contains information specific for that class. Essentially, subclasses inherit all methods and values the base class contains. A benefit of this is that you can use any subclasses as you would use a base class. A list of the base class can contain and subclasses despite being different classes.

        Activity[] activities = new Activity[]

        {

            new BreathingActivity(logger),

            new ReflectionActivity(logger),

            new ListingActivity(logger)

        };

This is an example of all subclasses being in a list. Because they also all share the base class’s methods,

 if (menuOptions[0].ToLower().Contains(userInput))

            {

                activities[0].RunActivity();

            }

            else if (menuOptions[1].ToLower().Contains(userInput))

            {

                activities[1].RunActivity();

            }

            else if (menuOptions[2].ToLower().Contains(userInput))

            {

                activities[2].RunActivity();

            }

            else if (menuOptions[3].ToLower().Contains(userInput))

            {

                chosenQuit = true;

            }

They can all be ran using the same method. The slight difference here is that RunActivity is an abstract method, meaning that each subclass overrides this method with its own. However, because it is contained in the base class, it still is a viable method between all subclasses. Using inheritance, code can be more easily organized and objects can be used in similar manners.