Inheritance is a way in object-oriented programming where a class inherits/gets the attributes and methods of a parent/base class.

Inheritance makes code cleaner as it makes it unnecessary to redeclare variables or use numerous getters and setters.

Inheritance can be used to keep an inventory of what stores sell. For example if a car company has several subsidiaries that each produce its own luxury brands of cars, but all the subsidiaries produce some base car brands exactly the same. The principle of Inheritance suggests that instead of having each subsidiary independently list the base model it sells, we create a parent class with a list of all the base models available and have each of the subsidiaries pull from that list. The subsidiaries will then only have to worry about creating lists the unique brands they offer.

protected void DisplayTimer(int moreTime = 0, bool WriteLine = true) {

        for (int i = 5 + moreTime; i > 0; i--) {

            Console.Write(i);

            Thread.Sleep(1000);

            Console.Write("\b \b");

            if (i > 9 || i < 0) {

                Console.Write("\b \b \b \b");

            }

        }

Above is an example from my code. The method above is going to be used by all child classes of this parent class. Instead of declaring this method in the Main class where it can be tampered with, I set it as protected and declare it in a base/parent class. All children of this class then have access to this method.