# Unit 4 - Programming Projects

Create a new project/solution for each of the following projects. Name the solution Unit?Project? and the project Project?.

Example:

* Solution: Unit1Project4
* Project: Project4

Some projects will require files from previous projects. Copy files from previous projects to the new projects using File Explorer in Windows. The files are then added to the project in Visual Studio.

## Events

### Project 1

Create a new console application. Modify the Stock class from a previous exercise and add the following:

* An event called CurrentPriceChanged that occurs when the Stock’s current price changes.

Create a test program that constructs an instance of Stock and subscribes to its CurrentPriceChanged event. Test the program by making the event take place and handling it by printing the Stock object. Also, try unsubscribing from the event.

## Project 2

Create a new console application. Modify the GeometricShape and Rectangle classes from a previous exercise and add the following:

GeometricShape Class

* An event called ColorChanged that occurs when the GeometricShape’s color changes.
* An event called FilledChanged that occurs when the GeometricShape’s fill changes.

Rectangle Class

* An event called HeightChanged that occurs when the Rectangle’s height changes.
* An event called WidthChanged that occurs when the Rectangle’s width changes.

Create a test program that constructs an instance of Rectangle and subscribes to all its events. Test the program by making the events take place and handling it by reporting objects new state. For example, when the Rectangle’s height changes print “Height is now 77”.

## Collections

### Project 3

Create a console application. Using the CardRank, CardSuit, and PlayingCard types coded in a previous exercise, create a deck (collection) of PlayingCards. There are several ways to do this…here is a hint: you can iterate through the values of an enumeration.

**Note**: A standard deck of playing cards has 52 cards. One playing card for each rank/suit combination.

#### **Extra Challenge**

1. Shuffle the deck of PlayingCards so they are in a random order.
2. Deal 5 PlayingCards to 3 players (each player is a collection). Ensure that when a card is dealt, it is no longer part of the deck.

### Project 4

Create an array that will store grades for 10 students. No user input is required.

Example: 40, 55, 30, 98, 57, 76, 82, 98, 55, 55

**Step 1** - Create a new array that contains the data from the first, but with the grades in reverse order. Do not use a literal array in your code. Obtain the data from the first array.

Example: 55, 55, 98, 82, 76, 57, 98, 30, 55, 40

**Step 2** - Print the each unique number in the first array. If there are duplicates of a number, do not print them.

Example: 40, 55, 30, 98, 57, 76, 82

**Step 3** - Print the number of occurrences of each grade value.

Example:

40 (1)

55 (3)

30 (1)

98 (2)

57 (1)

76 (1)

82 (1)

\*\*Step 4 \*\*- Sort the values in the array. Hint: Research a sorting algorithm and follow it.

Example: 30, 40, 55, 55, 55, 57, 76, 82, 98, 98

**Step 5** - Repeat steps 1-4, but use a List.

**Step 6** - Using a Dictionary, store the letter grade chart found in the course outline of this course. For example, to achieve the letter grade “A+” you must have a grade of 90 or higher.

**Step 7** - Iterate through the grades and print the grade and corresponding letter grade beside it. The letter grade will be determined using the Dictionary from step 6.

Example:

40 (F)

55 (D)

30 (F)

98 (A+)

57 (D)

76 (B+)

82 (A)

98 (A+)

55 (D)

55 (D)