# Unit 1 - Programming Projects

Create a new project/solution for each project.

**NOTE**: The files from these projects may be used and modified in future exercises.

## Classes and Objects

## Project 1

Create a new console application. Code a class named Rectangle to represent a rectangle.

The class contains:

* Two decimal data fields named width and height that specify the width and height of the Rectangle. The default values are 1 for both width and height.
* A no-arg constructor that creates a default rectangle.
* A constructor that creates a rectangle with the specified width and height.
* A method named GetWidth() that returns the width of the Rectangle.
* A method named GetHeight() that returns the height of the Rectangle.
* A method named GetArea() that returns the area of this rectangle.
* A method named GetPerimeter() that returns the perimeter.
* A ToString method that returns the Rectangle in the following format: W: {width} H: {height} A: {area} P: {perimeter}

Write a test program that creates two Rectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

## Project 2

Create a new console application. Code a class named Stock that contains:

* A string data field named symbol for the Stock’s symbol.
* A string data field named name for the Stock’s name.
* A decimal data field named previousClosingPrice that stores the Stock price for the previous day.
* A decimal data field named currentPrice that stores the stock price for the current time.
* A constructor that creates a Stock with the specified symbol, name, current price. The Stock’s previous closing price will initially equal the current price.
* A method name GetCurrentPrice() that returns the current price of the Stock.
* A method name SetCurrentPrice(currentPrice : decimal) that modifies the previous close price to the state of the current price, then updates the current price to the specified price.
* A method named GetChangePercent() that returns the percentage changed from previousClosingPrice to currentPrice. The formula is: price difference / previous closing price x 100. If the change percentage is a negative, this indicates an increase.

Write a test program that creates a Stock object with the stock symbol ORCL, the name Oracle Corporation, and the current price of 34.5. Set a new current price to 34.35 and display the price-change percentage.

## Enumerations

## Project 3

Create a new console application. Create an enumeration called LetterGrade with the following values.

* F
* D
* C
* CPlus
* B
* BPlus
* A
* APlus

Create a class called GradeItem. The class has the following members:

* A decimal data field named score for the GradeItem’s score.
* A decimal data field name maxScore for the GradeItem’s max score.
* A constructor that creates a GradeItem with the specified score and max score.
* A method named GetLetterGrade() which returns a LetterGrade type value. The method implementation will return the LetterGrade for the GradeItem’s score.
  + A+ >= 90%
  + A >= 80%
  + B+ >= 75%
  + B >= 70%
  + C+ >= 65%
  + C >= 60%
  + D >= 50%
  + F >= 0%

The program will do the following:

* Create an array that will hold 8 GradeItem objects. Each object should evaluate to one of the LetterGrades from the enumeration.
* Iterate through the array printing the LetterGrade for each GradeItem.