# Unit 4 - Exercises

Create a new Visual Studio solution called ADEV2005Unit4 containing a Console Application project called Unit4Exercises.

## Setup

1. Using the File Explorer application in Windows, locate the CardRank.cs, CardSuit.cs and PlayingCard.cs code files from the ADEV2005Unit2 solution.
2. Copy the three files listed in step 1.
3. Paste the three files into the root directory of the Unit4Exercises project.
4. In Visual Studio, right-click the Unit4Exercises project node in the Solution Explorer panel and choose the Add > Existing Item… menu item from the context menu.
5. In the dialog that appears, select the three files from the root directory of the Unit4Exercises project (it should default to this location).
6. Click the “Add” button.

Those three files are now part of the project. Note: Stopping at step three does not include those files in the project.

## Events

**EX1**. Declaring an event.

1. Declare an event called “Flipped” in the PlayingCard class. The PlayingCard class is in the PlayingCard.cs code file. This event will occur when the PlayingCard is flipped.

**EX2**. Create an “On” method for the event. The “On” method associated with an event implements what code takes place when the event is raised.

1. Declare a method called “OnFlipped” within the PlayingCard class. Declare all your “On” method together right above the ToString method. if the class does not have a ToString method, the “On” methods will be at the end of the class.
2. Code the implementation of the OnFlipped method such that it invokes the methods that are handling the event.

**EX3**. Raising the event.

1. Update the Flip method to raise the Flipped event.

**EX4**. Testing the event.

1. In the Program class, declare a new method called “PlayingCardFlipped” that will handle the Flipped event of a PlayingCard.
2. The implementation of the PlayingCardFlipped method will print “Playing card was flipped!” to the console.
3. In the Main method:
   1. Construct an instance of the PlayingCard class. Use either constructor and provide you own values.
   2. Print the instance created in step 1 to the console.
   3. Flip the playing card.
   4. Print ““Press any key to continue…” and pause the application with a call to ReadKey().
4. Run the program and analyze the results.

### Questions

1. Is the program’s output what you expected?
2. Did the event take place? Why or why not?

**EX5**. Subscribe to an object’s event.

1. In the Main method of the program, after flipping the instance of PlayingCard but before “Press any key to continue…”, add code to subscribe (attach handler method) to the instance’s Flipped event.
2. Run the program and analyze the results.

### Questions

1. Is the program’s output what you expected?
2. Did the event take place? Why or why not?

**EX6**. Update the program.

1. Move the subscription statement before flipping the card.
2. Run the program and analyze the results.

### Questions

1. Is the program’s output what you expected?
2. Did the event take place? Why or why not?
3. If a called to the instance Flip behaviour happened a second time after the first, what is the expected output? Does the event take place? If you think it does, how many times does it happen in the program?

## Collections

Before preceding with the following exercises, comment the code from the previous exercises.

**EX7**. Create an ArrayList.

1. Declare and instantiate an instance of an ArrayList with the identifier friends.
2. Add data to the ArrayList for three of your friends. Add the following data: name, if the friend is taller than you or not, and the data of their birth. For the date of birth use the structure DateTime. You can use the constructor DateTime(int, int, int) that initializes a new instance of the DateTime structure to the specified year, month, and day.
3. Print the data for each friend to the console on its own line. Use a for loop to accomplish this task.

**EX8**. Create a List of PlayingCards.

1. Declare and instantiate an instance of a List with the identifier cards.
2. Add three different PlayingCard objects to the List.
3. Print the PlayingCard objects to the console on its own line. Use a for each loop to accomplish this task.

**EX9**. Create a Dictionary of the courses you currently taking.

1. Declare and instantiate an instance of a Dictionary with the identifier courses.
2. Add data for the courses you’re currently taking to the Dictionary. The key for each element is the course code (ex. “ADEV-2005”) and the value is the name of the course (ex. “Programming 2”).
3. Print the course descriptions to the console on its own line. Do not use a loop to accomplish this task.