COMP123 – Programming 2

Final Exam - Practical
Pick Highest Card

Due Week #14 (August 17th, 2017) by end of class

Value 25%

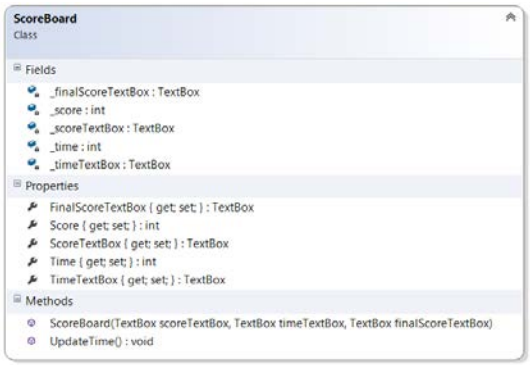
Pick Highest Card

Time Limit: 1 hours 30 minutes **Maximum Mark:** 42

Overview: Using the Project Template Provided, complete the simple game project "Pick Highest Card". In this game, the user clicks the **Deal Button** and Cards are displayed. The user can then attempt to pick the Highest Card Randomly. The Game will provide a hint after the first card is chosen. The user will gain points if the Highest Card is chosen earlier. You will complete the project by adding the missing code, where appropriate, any controls and forms that are missing.

Instructions :
(13 Marks: GUI, 20 Marks: Functionality, 5 Marks: Internal Documentation, 4 Marks: Version Control)

1. **SplashForm: This form needs to be created (SubTotal: 13 Marks: GUI, 4 Marks: Functionality)**
    a. Add a **new form** to the project and name it **SplashForm** (2 Marks: GUI).
    b. Set the **StartPosition** property to **CenterScreen** (1 Mark: GUI).
    c. Set the **BackColor** property to Black (1 Marks: GUI).
    d. Set the **FormBorderStyle** to **None** (1 Mark: GUI)
    e. Set the **ControlBox**, **MaximizeBox** and **MinimizeBox** properties to **False.** (3 Marks: GUI)
    f. Set the Form's **Size** property to **600**, **500** (1 Mark: GUI)
    g. Add a **Timer** control to the form from the Components section of the **ToolBox** (1 Mark: GUI)
    h. Set the Timer control's properties so that it is enabled with an interval of 3000 milliseconds (2 Marks: GUI).
    i. Create the Timer's **tick** event handler (1 Mark: GUI).
    j. Write code so that the **SplashForm** is hidden and the **PickHighestCardForm** is shown after the Timer's tick event is triggered. Ensure that this only happens once. (3 Marks: Functionality).
    k. In the **Program.cs** file change the **Application.Run** method to start the SplashForm first (1 Mark: Functionality).



2. **ScoreBoard class:** **This class needs to be created** (SubTotal: 11 Marks: Functionality)
    a. Use the class Diagram above to scaffold out the **ScoreBoard** class (5 Marks: Functionality)
    b. Ensure that the public **Score** property sets the **ScoreTextBox.Text** property and the **FinalScoreTextBox.Text** property to the **_score** value (2 Marks: Functionality)
    c. Ensure that the public **Time** property sets the **TimeTextBox.Text** property to the **_time** value (1 Mark: Functionality)
    d. The **UpdateTime** method will convert the current value of the **TimeTextBox.Text** property to an integer, subtract 1 from it, then assign it to the Time property. Then set the **TimeTextBox.Text** property to the **Time** value (which you will have to convert to a string) (3 Marks: Functionality)
3. **PickHighestCardForm**: **This form needs to be modified** (5 Marks: Functionality)
    a. Define a **_scoreboard** private instance variable and a linked **ScoreBoard** public property (1 Mark: Functionality).
    b. In the Form's **_reset()** method, set the **ScoreBoard.Score** to 0 and the **ScoreBoard.Time** to 30 (1 Mark: Functionality).
    c. In the Form's **PickHighestCardForm_Load** event handler method, instantiate the ScoreBoard object and pass in a reference to the **ScoreTextBox**, **TimeTextBox** and **FinalScoreTextBox** (1 Mark: Functionality)
    d. In the **_showUserMessage** method, uncomment the **ScoreBoard** reference (1 Mark: Functionality).
    e. In the **CountDownTimer_Tick** event handler method, uncomment the **ScoreBoard** reference (1 Mark: Functionality).
4. Include **Internal Documentation** for your Application **(5 Marks: Internal Documentation):**
    a. Ensure you include a **comment header** for your **C# files** that indicate: the **App name**, **Author's name**, **App Creation Date**, **App description** (2 Marks: Internal Documentation).
    b. Ensure you include a **section headers** for all of your **Event Handlers, Classes,** and any **functions** (1 Marks: Internal Documentation)
    c. Ensure all your code uses **contextual variable names** that help make the files human-readable (1 Marks: Internal Documentation).
    d. Ensure you include **inline comments** that describe your GUI Design and Functionality. (1 Mark: Internal Documentation).
5. Share your files on **GitHub** to demonstrate Version Control Best Practices **(4 Marks: Version Control).**
    a. Your repository must include **your code** and be well structured (2 Marks: Version Control).

b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

BONUS SECTION

Make the sounds work for dealing and flipping cards (located in the resources folder) – 5 Bonus Marks

SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your project files
2. A link to your project files on GitHub.

Feature	Description	Marks
GUI / Interface Design	UI Controls meet the application requirements. Display elements are deployed in an attractive manner. Appropriate contrast is applied to application UI Controls and any background colours applied so that all text is legible.	13
Functionality	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	20
Internal Documentation	A program header is present and includes the name of the program, the name of the student, student number, date last modified, a short revision history and a short description of the program. All methods and classes include headers that describe their functionality and scope and follow commenting best practices. Inline comments are used to indicate code function where appropriate. Variable names are contextual wherever possible.	5
Version Control	GitHub commit history demonstrating regular updates.	4
Total		**42**

EVALUATION

This exam is weighted **25%** of your total mark for this course.