

CIS 200 (Scholars): Project 9 (40 points)

Due Monday, Nov 27th by 11:59pm

Reminder: Programs submitted after the due date/time will be penalized 10% for each day the project is late (not accepted after 3 days, i.e. Midnight, Thu, Nov 30th)

Reminder: As stated in the *Academic Integrity Policy* in the syllabus for this course, unless otherwise stated, **ALL projects are intended to be done individually. Do NOT share your code with anyone.** If it is not your work, don't submit it as yours. Refer to the policy within the course syllabus which states, ***“If two (or more) students are involved in ANY violation of this policy, at a minimum, ALL students involved receive a zero for the assignment and the offense is officially reported to the KSU Honor Council. The second offense results in a failing grade for the course and possible suspension from the university (this decision is made by the K-State Honor Council).”***

Note: If you use code that YOU did not develop (i.e. from another text or found it on the internet) **you must cite your source** in a comment above the line(s) of code that is not your own work, otherwise it will be considered **plagiarism** (*representing others' work, whether copyrighted or not, as one's own*).

Assignment Description:

To demo the similarities between *Java* and *C#*, create the following MVC **console** application in *C#* -

Create a **model** class (**Poker_Hand**) to hold the contents of a single poker hand (see **poker.txt**) with a *ToString* method used to display the contents of a single hand. **Controller (Prog9)** will use an *ArrayList* of *model* objects to hold an undefined number of different poker hands, reading in the hands from **poker.txt**. (Don't assume a 1,000 – read until EOF).

Create a **View** class used to display the following calculated results (calculation done in the controller). In addition, have *View* class write all of the poker hands contained in the *ArrayList* to a file called **poker_results.txt**, using the *ToString* method defined in the model class, followed by the results shown below. (You can also *append* to the existing *poker.txt* file but will have to rename the file afterwards to *poker_results.txt*)

Results (---- hands):	(display how many hands were read in)
Royal flush: -.-%	(% of each, decimal to 1 decimal)
Straight flush: -.-%	
Four of a kind: -.-%	
Full house: -.-%	
Flush: -.-%	
Straight: -.-%	
Three of a kind: -.-%	
Two pairs: -.-%	
One pair: -.-%	
Other: -.-%	

If you are unfamiliar with the possible “hands” of poker, see Project 2 or go to:
<https://www.pokerstars.com/poker/games/rules/hand-rankings/>

Implementation Requirements:

Your project will contain **three** classes. No I/O in the *model* or *controller* classes. *View* class will handle console based IO only (no GUI). Your *controller* class will contain **ONLY** a Main method (and methods that are not a required part of the *Model* or *View* class) and will **NOT** contain any *Input* or *Output* statements.

Documentation: Put a description of the project at the top of the file **and at the top of each method**.

Please use this template for the [top of the file](#):

```
/** (description of the project)
 * @author (your name)
 * @version (which number project this is)
 */
```

Please use this template for the [top of each method](#):

```
/** (description of the method)
 * @param (describe first parameter)
 * @param (describe second parameter)
 * (list all parameters, one per line)
 * @return (describe what is being returned)
 */
```

Requirements

Important: ALL files must compile at the command-line with the statement: **csc filename.cs**. Project must run at the command-line with the statement: **Proj9**. **Make sure and test this before submitting**. Submit ALL needed files and submit the *CORRECT* files (i.e. the ones that compile). It is very important that you learn to submit what is needed to your *client*.

Submission: Read these instructions carefully or you may lose points / get a ZERO on your project

<p>Programs that do not compile will receive a grade of ZERO, regardless of the simplicity or complexity of the error, so make sure you submit the <i>correct</i> file that <i>properly compiles</i>.</p>
--

If you used **Visual Studio** to create and compile your project, simply zip the entire project folder, rename as **Proj9.zip** and submit.

If you used a **text editor and command line compiler** to create and compile your project, first create a folder called **Proj9**, then copy your *three* completed .cs files and your **Proj9.exe** file into that folder. Then, right-click on that folder and select “**Send To → Compressed (zipped) folder**”. This will create the file **Proj9.zip**.

Log-in to Canvas and upload your **Proj9.zip** file. **Only a .zip file will be accepted for this assignment in Canvas**. Put your full name and Project 9 in the comments box.

*Important: It is the **student's responsibility** to verify that the **correct** file is **properly** submitted. If you don't properly submit the correct file, it will not be accepted after the 3-day late period. **No exceptions**.*

Grading: Programs that do not compile will receive a grade of 0.

Those submissions missing the exe file will receive ½ credit maximum.

Requirement	Points
Proj9 class properly converted to C#. Contains NO write, data validation or input statements / Code + Execution:	24
• Properly creates objects of the Poker_Hand and View classes.	
• Contains only a Main method (other supporting methods are OK)	
• Reads in ALL poker hands in the text file until EOF is reached (1,000 not hardcoded)	
• Properly creates object(s) of the <i>model</i> class and stores in an ArrayList	
• Correctly calculates and displays the results as shown (% to one decimal)	
• Correctly writes hands stored in ArrayList to results file uses ToString method defined in the <i>model</i> class, using the View object	
• Correctly writes summary results to the end of the results file using the View object	
• Documentation Correct / Updated	
Poker_Hand (model) class	5
• Correctly holds the contents of a single Poker hand	
• Properly defines a <i>ToString</i> method	
• No I/O done in this class	
• Documentation Correct / Updated	
View class properly created in C#. Contains ONLY the <i>Input-Output</i> portion of the program. Documentation Correctly Added	10
Proper Submission with full name & project # in Comment box. Correct Filenames	1
Minus Late Penalty (10% per day)	
Total	40