

Final Project Statement of Work Template

Group Number:

3

Group Leader:

Violet Yousif

Names of Group Members:

Violet Yousif, Justin Smith, Nguyen Tran

Summary of what your program will do (be very specific):

We are going to be using the Deck of Cards API found here to create a couple of mini games that will implement each requirement for the final project. The numbers in parenthesis proceeding each idea indicates which rubric requirements will be met. First, we will create a parent class that will contain the struct for each individual card (struct members for suits and value), and we will set up virtual functions for the child classes to overload and overload operators for printing the results from the structs. (1, 2, 3, 4).

In game number one, inside of an inherited class, the user will be guessing a random card value. If they don't guess it on the first try, the program will return hints "higher" or "lower" to guide the user in the direction of the answer (4, 8, 9, 11, 12).

The second inherited class will ask the user to draw as many cards as they would like from a single deck, and then sort them in order by either suit or numeric value. If the user is dealt a joker, which would be found via searching and exception handling, then a message would be returned, and that card would be replaced. (4, 5, 6, 7, 8, 9, 11, 12).

To expand on the above, exception handling will be implemented in each class by ensuring that the data from the API is properly parsed into the data vector and the struct members (i.e. null values). It will also be used by ensuring that the user only gets 5 guesses in game number one. (10).

Note: Comments (11) and building the Repo (14) should be finished after getting the program running.

Decomposition with Assignments:

Use the space to list the decomposition or outline of your program. Next to each section, provide the name of the student who will be working on that portion.

	Decomposition	Responsibility
1	3 User-Defined Classes	V.Y, J.S, N.T
2	Class Inheritance w/ Polymorphism (function overloading)	J.S
3	Class Composition (class vars w/ a 'has-a' relationship)	J.S
4	2 Overloaded Operators	V.Y
5	Definition and Use of a template	N.T
6	Function that performs Searching	V.Y
7	Function that performs Sorting	N.T
8	Use of a Vector object in the iteration of its contents	J.S
9	Dynamic Memory Allocation & Deletion	N.T
10	At least one use of exception handling	V.Y
11	Reading and Writing of data to a file (or API?)	J.S
12	I/O to/from console	J.S
13	Well-Written Comments	V.Y, J.S, N.T
14	Fully updated GitHub Repo	V.Y, J.S, N.T