**Assignment #0** (*welcome* *assignment*)

Problem Solving and Programming in C++

Department of Computer Science

Old Dominion University

**Objectives:**

The main objective of this assignment is to help you review some of the concepts related to functions in C++.

**General Instructions:**

1. Read the Background and Task Description below.
2. In this assignment, you will be given a working program.
3. Split the main function given into multiple functions.

**Task Description:**

You have been given a working program that outputs a shuffled deck of cards. A deck of cards has 4 suits where each suit has 13 cards ranging from Ace, King, Queen, Jack, 10…, 2. There are games using fewer than 13 cards for each suit, such as Belote. Thus, the program prompts the user to select the number of individual cards in a suit, where the minimum is five, and the maximum is 13.

The program will create a shuffled deck, test to see whether the deck has four of each card and then print the deck to the screen and a file (user will insert the filename) if the test is successful.

All functionality has been included in main, causing code segments to be repeated as well as diminishing the readability.

Rewrite the program by grouping calculations and related operations into functions. In particular, your program should include the following functions.

* **PrintList – DONE MAYBE**: This is a void function, that accepts an array, the size of the array, and a string to print the list, such as “Sorted Deck” or “Sorted Integers”, as parameters. Create two functions, one that accepts an array of integers, and another that accepts an array of strings.
* **FilePrintList – DONE MAYBE**: This void function, taking an array as parameter and the size of the array, and prints the array to a file, after prompting the user for a file name. Similar to the “PrintList” function, make two variations of the function, one that accepts an integer array, and another that accepts a string array.
* **RandomInt - DONE**: This is an int function (returns a random integer), that takes as a parameter an integer called ceiling. This function generates a random number between 0 and “ceiling -1”. It is used in the Deck Creation process and the random integer represents an index.
* **GetSize - DONE**: This is an int function, taking no parameters. It prompts the user to determine the number of individual cards in a suit. It should only accept integers in the range [5,13]. If the user enters a value outside of this range, the program should prompt the user to re-enter until a value in the appropriate range is inserted.
* **Menu**: A void function taking no parameters. It allows the user to select whether he wants to create another deck of cards or exit. It checks whether the user has inserted an invalid action. Allowed actions are (1: create deck, 0: exit).
* **Map – DONE** : This is a void function, taking an integer array as parameter, a string array, and an integer representing the size of the card deck. This function converts the contents of the integer array to strings and stores them in the string array. For example, convert 0 to “1”, or 1 to “2”, 10 to “jack”.
* **TestDeck - DONE**: This bool function goes through the generated deck before its conversion to a string array, and checks if all numbers appear exactly four times. It takes an integer array, and size of the array, as parameters. It returns true if test is passed, false otherwise.
* **CreateDeck**: This is a void function that encompasses the entire deck creation process, except prompting the user for the number of individual cards. It takes a string array as parameter and user’s select size of the suit (GetSize() output). This function is called when the user selects “action 1” from the menu.

Hints:

* Functions can be called from other functions
* You can pass an array to a function in the following manner:
  + Function Definition: void foo(int arr[], int size)
  + Array Definition: int array[13]
  + Function call: foo (array, 13)
* Function foo with various return types
* int foo (int arr[], int size)
* string foo (int arr[], int size)
* double foo (int arr[], int size)
* You can store the return value in a variable, or output it
* int x = foo(arr, 5)
* std::cout<<foo(arr, 5)
* There needs to be a return statement inside the function, if there is a return type (Figure 1).

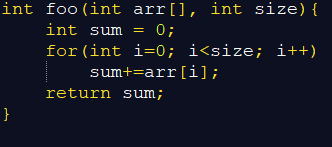


Figure . Function with integer return type

As you introduce each function, replace the code in main() by the appropriate function call. Keep in mind that for some functions, any changes that occur within the function body must also be visible in main.

This is a working code, which should not be altered. You can only restructure the code. Splitting the current program into multiple functions makes it modular and easier to understand.

**Submission notes for Task:**

* Zip the entire Code::Blocks project containing all the .cpp, .h, .cbp files and name the zipped file “Assg0\_cslogin.zip”, where the cslogin is your login ID for the computers at the Department of Computer Science at ODU. Submission with a different file name other than Assg0\_cslogin will receive -10 points off your final points for the assignment.

* Submit the zipped file using the appropriate Blackboard link.

* Using global variables will cause a 10 point deduction to your final score.

**Sample Output:**

