# Programming Assignment #1

**Due: 10/02/2022 23:59:59**

**Topic:** printing a given number of cards from a deck in a nice format.

**Objective:** review string, array, and pointer in C and practice new features in C++.

**Description:**

You are asked to draw poker cards from a deck of 52 cards and print them to the screen in a nice format. The input to the program is the number of cards to be printed, and the output is a graphical representation of the cards on a regular ASCII terminal. Each card is randomly drawn from a deck of 52 cards ordered by pip, (i.e. 1S 1H 1D 1C 2S 2H 2D 2C .....). The output of the cards will be formatted such that up to 5 cards are arranged in a row. Extra cards are automatically wrapped over to the next row. A drawn card should not reappear. Therefore, the number of cards that a user asks to display should not exceed 52. If it does, you should print an error message such as *"Sorry, number of cards can not exceed 52."*

**Assignment directory:** `/usr/local/class/oop/assign/assign1`

**Sample output:** a sample output is given as follows.

```
mtchi@oop 10:34am [86] assign1> CardTest_x64-sample
Usage: CardTest_x64-sample NCards [Seed]
mtchi@oop 10:34am [87] assign1> CardTest_x64-sample 3
********** ********** **********
*6D      * *8C      * *1S      *
*        * *        * *        *
*  D  D  * *  C  C  * *        *
*        * *    C   * *        *
*  D  D  * *  C  C  * *    S   *
*        * *    C   * *        *
*  D  D  * *  C  C  * *        *
*        * *        * *        *
*       6* *       8* *       1*
********** ********** **********
ID: u1234567

mtchi@oop 10:35am [89] assign1> CardTest_x64-sample 6 123
********** ********** ********** ********** **********
*10S     * *10H     * *9D      * *5C      * *2S      *
*  S  S  * *  H  H  * *  D  D  * *        * *        *
*    S   * *    H   * *        * *  C  C  * *    S   *
*  S  S  * *  H  H  * *  D  D  * *        * *        *
*        * *        * *    D   * *    C   * *        *
*  S  S  * *  H  H  * *  D  D  * *        * *        *
*    S   * *    H   * *        * *  C  C  * *    S   *
*  S  S  * *  H  H  * *  D  D  * *        * *        *
*      10* *      10* *       9* *       5* *       2*
********** ********** ********** ********** **********
**********
*7H       *
*        *
*  H  H  *
*        *
*  H H H *
*        *
*  H  H  *
```

```
*           *
*          7*
***********
```

**What are given:**

You are given the following files to start with:

- a sample Makefile: `Makefile`
- a partial testing file: `CardTest.cc`
- a sample header file: `Cards.h`
- a pair of files for printing nice looking string on an ANSI-enabled terminal: `AnsiPrint.cc` and `AnsiPrint.h`. (See the comments in the files for how to use the given functions to print a string with special ANSI control characters.)

**What to hand in:**

You are asked to write two C++ source files: `CardTest.cc` and `Cards.cc`. If you name your source files differently, remember to modify the Makefile accordingly and hand it in with other source files. A sample executable program compiled for the server (`CardTest_x64-sample`) is also included in the assignment directory for your reference. You must submit all program source code electronically. (See the 2nd handout for how to do that.)

**Implementation details:**

You can feel free to choose the platform on which you develop your program. If you choose the PC platform, you may not want to use the `AnsiPrint()` function at the development stage since you can not see color on your screen without loading extra drivers. However, you can add these functions on UNIX after you have finished developing other functions. To test these color-printing functions, you must be on an ANSI-enabled color terminal.

In this assignment, you are asked to use as many features of C++ as you learn in the class. You should be able to gain some experience by studying the given source code. In addition, you are encouraged to use as many C++ features as possible when writing your code. For example,

- C++ iostream functions for console I/O
- `const` modifier for function parameters and constants
- references
- default function parameters

**Notes:**

In general, *late* homework may receive fewer points than *incomplete* homework. DO comment and nicely format your code to avoid any penalty imposed by the grader simply because he/she can not easily understand what you intend to do. I will encourage the grader to pretend to be a normal reader/programmer instead of an expert programmer in grading your homework assignments.

You can submit your homework electronically at any time. A later submission will overwrite earlier ones and only the date of your last submission counts. The grader will start grading your homework sometime after the deadline, and your homework will be graded only ONCE unless a special arrangement is made in advance.