

# CS 180: Data Structures, Fall 2013

## Homework 1

Due *via email* by 11:59pm on Saturday, Sept. 7

This is a homework you must complete individually. Each problem here is focused on learning the basic syntax of C++. You should submit a separate cpp file for each problem, commented with your name and the problem number at the top (as well as other comments throughout which describe the program as necessary). Note that these should be tested before submitting them; make sure they actually compile and work!

Please type all answers and email to the instructor at echambe5 - at - slu.edu by 11:59pm on the date due.

1. Write a short C++ program that presents the user with a choice of your 5 favorite beverages (tea, coffee, Coke, water, or whatever). Then prompt the user to choose a beverage by entering a number 1-5. Output which beverage they chose. If they enter a number other than 1-5, output "Error: Choice was not valid."

Note: You can assume that the user will enter a value integer, so you don't have to deal with error handling if they enter something totally crazy.

2. Write a function `printSquare` which takes as input an integer `size` and a character `fillChar`, and prints a square of the specified size formed from that character. For example, if `size` is 4 and `fillChar` is '@, then it should print:

```
 @@@@  
 @@@@  
 @@@@  
 @@@@
```

Write a (short) main which calls this function at least twice with different inputs to verify that it works correctly.

3. Write a short C++ function called `allDistinct` that takes as input a pointer to an array of int values and an integer which is the size of the input array, and then returns true if numbers are all different from each other (and false otherwise). Then write a main which tests your function.

Note: to test for correctness, at a bare minimum you need several different arrays of different sizes, some of which should return true and some of which should return false. You also should output appropriate descriptions of the test, such as what the array is and what the return value is, so that when running it I can clearly see if it is working or not.

4. Extra Credit:

Write an efficient C++ function that takes any integer value  $i$  and returns  $2^i$  as a long value. Your function should *not* multiply 2 by itself  $i$  times; there are much faster ways of computing  $2^i$ .

Note: Faster solutions will receive more extra credit.