

# CS261: HOMEWORK 2

## Due 04/24/2017 by noon (12pm)

Submit two files via the TEACH website:

[https://secure.engr.oregonstate.edu:8000/teach.php?type=want\\_auth](https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth)

### General Instructions

For this homework you are given a skeleton code in the following files: `dynArray.h`, `dynArray.c`, `stackapp.c`, and `makefile`. In `dynArray.c` and `stackapp.c`, you are supposed to complete certain functions or their parts, **without changing other content**. Complete and submit `dynArray.c` and `stackapp.c`. You must use the same names of files, functions, and ALL variables, we specified for you in the skeleton code. This is because our grading will be based on a script that assumes that you use the same names. Make sure you use the provided `makefile` to compile your code on the ENGR Unix server. Note zero tolerance for compiling errors.

### Part 1 – 70 points : Implementation of the Dynamic Array, Stack, and Bag

Complete the implementation of the following seven functions in `dynArray.c`. The comments for each function will help you understand its purpose. DO NOT change the provided header file `dynArray.h`, and DO NOT change the other completed functions in `dynArray.c`.

#### Scoring:

- 1) `void _dynArrSetCapacity(DynArr *v, int newCap)` = 10 points
- 2) `void addDynArr(DynArr *v, TYPE val)` = 10 points
- 3) `void removeAtDynArr(DynArr *v, int idx)` = 10 points
- 4) `void removeDynArr(DynArr *v, TYPE val)` = 10 points
- 5) `void pushDynArr(DynArr *v, TYPE val)` = 10 points
- 6) `void popDynArr(DynArr *v)` = 10 points
- 7) `int containsDynArr(DynArr *v, TYPE val)` = 10 points

## Part 2 – 30 points : Application of Stacks

Stacks can be used to check whether an expression has balanced parentheses ( ), braces { }, and brackets [ ]. For example, a balanced expression is

$$\{(x + y), [x + (y + z)]\}$$

and an unbalanced expression is

$$\{(x + y), [x + (y + z)]\}.$$

Write a function, `int isBalanced(char* s)`, in file `stackapp.c` that solves this problem using a stack. Alternative solutions that produce correct results BUT DO NOT USE a stack will not receive credit.

The function `isBalanced(char* s)` should read through the string using the function `nextChar(char* s)`, which has already been implemented for you. Then, apply a stack to do the test. It should return either '1' for true, or '0' for false. Note that for this task you would need to make the following change in file `dynArray.h`:

Change `#define TYPE double` to `#define TYPE char`

### Scoring:

1) `int isBalanced(char* s)` = 30 points