1 Structs and Pointers
Recall that direct access to members of a struct uses DOT notation. One can also use
t Pointers to point to structs. For example,

```c
typedef struct {
    int xpos;
    int ypos;
} Point;

Point *pt;
(*pt).xpos = 0;
(*pt).ypos = 1;
```

The brackets are necessary; thus there is a common shorthand: `pt->xpos`

Note that one often ends up with structs dynamically allocated. The function
`malloc` is similar to `calloc`, except that it takes only one argument, the number of bytes
needed.

```c
Point *Q;
Q = malloc( sizeof(Point) );
```

2 Example Code: student.c

```c
#include<stdio.h>
#include<stdlib.h>

#define MAX_STUD 25
#define MAX_LEN 20

typedef struct
{
    char firstName[MAX_LEN];
    char lastName[MAX_LEN];
    float grade;
} Student;
```
typedef struct
{
    int numStuds;
    Student stud[MAX_STUD];
    float average;
} Class;

void readClass(Class*);
void writeClass(Class*);

int main()
{
    Class *r;
    r = malloc(sizeof(Class));
    readClass(r);
    writeClass(r);
    return 0;
}

void readClass(Class *C)
{
    scanf("%d",&C->numStuds);
    int s;
    float total=0.0;
    for(s=0; s<C->numStuds; s++) {
        scanf("%s%s%f", &C->stud[s].firstName, &C->stud[s].lastName,
             &C->stud[s].grade);
        total += C->stud[s].grade;
    }
    C->average = total/C->numStuds;
}

void writeClass(Class *C)
{
    int s;
    for(s=0; s<C->numStuds; s++) {
        printf("%s %s got %.2f\n", C->stud[s].firstName, C->stud[s].lastName,
                                           C->stud[s].grade);
    }
    printf("Class average is %.2f\n",C->average);
}