```
/* guitar.c
 * Christopher Brant
 * cbrant
 * ECE 2230
 * Section 001
 * Spring 2017
 * Programming Assignment #2
 * Due on 2/15/17 at 11:59 PM
 * Professor Walt Ligon
 */

#include <stdio.h>
#include <stdlib.h>
#include "guitar.h"

#define MAKEMODSIZE 20
#define SUBMODSIZE 15
#define FINISHSIZE 25
#define TYPELINE 3

 /* Allocate a new struct guitar_s and return a pointer to it */
guitar_t guitar_init(void)
{
        guitar_t new_guitar = (guitar_t)malloc(sizeof(struct guitar_s));

        return new_guitar;
}

/* Read from user all fields for a guitardb_guitar */
int guitarfill(guitar_t gpoint)
{
        int fields_read = 0;
        char line[MAKEMODSIZE];
        char sub[SUBMODSIZE];
        char fin[FINISHSIZE];
        char types[TYPELINE];

        // Everything below this is used to grab input from the user
        printf("Enter guitar make: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%s", gpoint->make);

        printf("\nEnter guitar model: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%s", gpoint->model);

        printf("\nEnter guitar submodel: ");
        fgets(sub, sizeof(sub), stdin);
        sscanf(sub, "%s", gpoint->submodel);

        printf("\nEnter guitar body type.\n");
        printf("Choose from:\nS (Solid)\nA (Arch-Top Hollow)\nT (Semi-Hollow)\n");
        printf("Enter here: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%c", &types[0]);

        gpoint->gtype = (guitartype_t)types[0];

        printf("\nEnter manufacture year: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%d", &(gpoint->year));

        printf("\nEnter finish description: ");
```

```
        fgets(fin, sizeof(fin), stdin);
        sscanf(fin, "%s", gpoint->finish);

        printf("\nEnter number of strings, a space, then number of pickups: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%d %d", &(gpoint->strings), &(gpoint->pickups));

        printf("\nEnter the neck, middle, and bridge pickup types separated by spaces.");
        printf("\nChoose from\nH (Humbucker)\nC (Single Coil)\nP (P90)\n");
        printf("F (Filtertron)\nN (None)\n");
        printf("Enter here: ");
        fgets(line, sizeof(line), stdin);
        sscanf(line, "%c %c %c", &types[0], &types[1], &types[2]);

        gpoint->neck = (pickup_t)types[0];
        gpoint->middle = (pickup_t)types[1];
        gpoint->bridge = (pickup_t)types[2];

        fields_read = 1;

        return fields_read;
}

/* Set the ID of a guitardb_guitar */
key_t guitar_setid(guitar_t gpoint, key_t keynum)
{
        int success = 1;

        gpoint->id_number = keynum;

        if (gpoint->id_number != keynum)
                success = 0;

        return success;
}

/* Get the ID of a guitardb_guitar */
key_t guitar_getid(guitar_t gpoint)
{
        key_t keynum = gpoint->id_number;

        return keynum;
}

/* Switch statement for pickups only used in guitar_print */
static void pickup_print(pickup_t ptype)
{
        switch (ptype)
        {
                case H :
                        printf("Humbucker\n");
                        break;
                case C :
                        printf("Single Coil\n");
                        break;
                case P :
                        printf("P90\n");
                        break;
                case F :
                        printf("Filtertron\n");
                        break;
                case N :
                default :
```

```
                        printf("None\n");

        }
}

/* Prints a guitar */
void guitar_print(guitar_t gpoint)
{
        printf("Guitar %d\n", gpoint->id_number);
        printf("Make: %s\n", gpoint->make);
        printf("Model: %s\n", gpoint->model);
        printf("Submodel: %s\n", gpoint->submodel);

        printf("Guitar body type: ");

        switch(gpoint->gtype)
        {
                case S :
                        printf("Solid Body\n");
                        break;
                case A :
                        printf("Arch-Top Hollow Body\n");
                        break;
                case T :
                        printf("Semi Hollow (thinline)\n");
                        break;
                default :
                        printf("None specified\n");
        }

        printf("Manufacture Year: %d\n", gpoint->year);
        printf("Finish description: %s\n", gpoint->finish);
        printf("Number of strings: %d\n", gpoint->strings);
        printf("Number of pickups: %d\n", gpoint->pickups);

        // Same thing with the enums here
        printf("Neck pickup type: ");
        pickup_print(gpoint->neck);
        printf("Middle pickup type: ");
        pickup_print(gpoint->middle);
        printf("Bridge pickup type: ");
        pickup_print(gpoint->bridge);
        printf("\n\n");
}

/* Returns 0 if equal, <0 if less than, >0 if greater than*/
int guitar_compare(guitar_t dbcursor, guitar_t findme)
{
        int match;

        if (dbcursor->id_number == findme->id_number)
                match = 0;
        else if (dbcursor->id_number > findme->id_number)
                match = -1;
        else if (dbcursor->id_number < findme->id_number)
                match = 1;

        return match;
}

/* Frees a guitar from memory */
void guitar_free(guitar_t gpoint)
{
```

```
        free(gpoint);
        gpoint = NULL;
}
```