8. PROGRAMMING ASSIGNMENT 8: UPDATED 12/2/17, 12/4/17

Read: Chapter 9
Programming: Name your program p8.c
DUE: THURSDAY, December 7, 2017 @ 6am

a) Your program will write a campus cash card online application for your campus cash card account using functions and a struct. main() will call all functions to read input and display the data. You will use a structure to represent the campus cash card account. The structure will look like the following. Use the defined symbolic constants. Include <string.h> for strcmp() and <stdlib.h> for exit().

NOTE: Type in the boxed code below, at the top of your program to allow the user to enter their name and account number into the initialized cash card account with a beginning account balance of $100.00

```c
#include <stdio.h>
define ASIZE 40
define MIN_BAL 0
define MIN_AMT 3.00
define MIN_DEPOSIT 100.00

struct campusCash
{
    double bal;
    char name[ASIZE];
    char acctNum[ASIZE];
};

// FUNCTION DECLARATIONS
int passcode(char *s);
void menu(void);
double deposit(void);
double dine (struct campusCash cc);
double kayak (struct campusCash cc);
void prtAcct (struct campusCash cc);

int main( void )
{
    int choice; // CampusCash menu choice
    int isInvalid = 1; // Invalid passcode
    struct campusCash cc1 = {MIN_DEPOSIT}; // Init $100 ccCard acct

    printf( "Welcome to Campus Cash! \n\n" );
    printf(" Enter Name and Account Number: ");
    scanf("%s", cc1.name, cc1.acctNum);
    do
    {
        isInvalid = passcode( cc1.name );
        if( isInvalid )
            printf("ERROR! Enter valid passcode. ");
    }while( isInvalid ); // Repeat until valid passcode

    prtAcct(cc1);
    do
    {
        menu(); // Display campus Cash menu
        printf(" Enter 1, 2, 3, 4, or 5: ");
        scanf("%d", &choice);
        getchar(); // Read newline
        switch(choice)
        {
            case 1:
                cc1.bal += deposit(); // Add to cc1 balance
                prtAcct(cc1); // Display balance
                break;
            ...
        }
    }while(choice != 5); // Repeat until exit
}"
```
b) The program will have the following function declarations:

```c
void menu( void );
int passcode( char *s );
void email (struct campusCash cc);
double deposit (void);
double dine (struct campusCash cc);
double kayak (struct campusCash cc);
void prtAcct (struct campusCash cc);
```

c) **main()** will define one variable, `cc1`, type struct campusCash:

```c
"struct campusCash cc1 = {MIN_DEPOSIT};  // Init $100 cc1 acct"
```

d) **passcode()** is called from **main()** and tests input string to match the exact magic passcode of “Cisfun”. **passcode()** returns a 1 if invalid password, zero if a valid password. Note: `strcmp()` on DevC++ returns a zero if both parameters are equal (No difference). Hint: use `scanf()` for input string.

e) **menu()** is called from **main()** and uses a series of printf() to display a menu of choices. e.g. printf("1: Deposit \n");

f) Write a function, **prtAcct()**, to display the data stored in the struct, see sample output below.

g) **deposit()** is called from **main()** and allows the user to enter a real number to be deposited into `cc1` account. Check for valid input greater than a minimum deposit amount of a hundred dollars. Loop and print an error message until a valid value is input. Return the deposit amount to **main()** to be added to the balance.

```c
double deposit()
{
    double amt;  // Amount of deposit
    do
    {
        printf("Enter deposit amount: ");
        scanf("%lf", &amt);
        getchar();
        if(amt<MIN_DEPOSIT)
            ...
    }while(amt<MIN_DEPOSIT);
    return amt;  // Return deposit amount
}
```

h) Repeat step f) for “**dine()**” method to subtract the amount from the `cc1` account. Return this meal price amount and deduct from the balance in **main()**. Pass in the struct `cc1` from **main()**.

**Validate in this order:**

1) if input is less than the minimum amount of 3, print the lower bound error message.
2) else if input is greater than the maximum amount of 100, print the upper bound error message.
3) else if input is more than current balance, print the insufficient funds error message.
4) if balance is less than 3, return 0
5) repeat asking for input/validation on error.

i) Repeat step h) for “**kayak()**”. Error check ranges are the same as **dine()**. These symbolic constants will help for a 1 day kayak rental of $23.00 and the required deposit amount of $575.00.

```
#define MIN_AMT_KAYAK 23.00
#define DEPOSIT_KAYAK 575.00
```

See sample output. [https://recreation.ucsd.edu/trips/kayaking/](https://recreation.ucsd.edu/trips/kayaking/)

j) Allow the user to email a transaction receipt with the user friendly message:

“Email of transaction sent to kevin@ucsd.edu”, using the input name in lowercase letters and `ucsd.edu` email address. Do not modify the original user name.

Hint: Use a local string, `strcpy()`, and `tolower()`.

Note: No actual email is sent, only display a message.
k) Allow your program to repeat this calculation as often as the user wishes until the user types in number “5” to quit the program.
   e.g. printf("ERROR - Enter again - 1, 2, 3, 4, or 5: " );

Display a friendly exit message. Use “exit(0);” for program termination.

**PA#8 SAMPLE OUTPUT:** (SAMPLE INPUT is in **bold** type below)

```
Welcome to Campus Cash!

Enter Name and Account Number: **Seth 123**
Hello Seth. Please enter your passcode: **FUN**
ERROR! Enter valid passcode.
Hello Seth. Please enter your passcode: **Cisfun**

Seth's Account#:123 has a balance of $100.00

******** CAMPUS CASH CARD ********
1: Deposit
2: Dine
3: Kayak
4: Email receipt
5: Exit

Enter 1, 2, 3, 4, or 5: **2**

Enter price of meal: **222.22**
ERR: Maximum meal is $100.00 !!

Enter price of meal: **90**

Seth's Account#:123 has a balance of $10.00

******** CAMPUS CASH CARD ********
1: Deposit
2: Dine
3: Kayak
4: Email receipt
5: Exit

Enter 1, 2, 3, 4, or 5: **1**

Enter deposit amount: **600.00**

Seth's Account#:123 has a balance of $610.00

******** CAMPUS CASH CARD ********
1: Deposit
2: Dine
3: Kayak
4: Email receipt
5: Exit

Enter 1, 2, 3, 4, or 5: **3**

Enter price of kayak rental plus deposit: **666.66**
ERR: INSUFFICIENT funds to kayak on balance of $610.00

Enter price of kayak rental plus deposit: **33.33**
ERR: Minimum rental+deposit is $598.00 !!

Enter price of kayak rental plus deposit: **598.00**

Seth's Account#:123 has a balance of $12.00
```

(continued on next page)
PA#8 SAMPLE OUTPUT: (continued)

(SAMPLE INPUT is in bold type below)

```
********** CAMPUS CASH CARD **********
 1: Deposit
 2: Dine
 3: Kayak
 4: Email receipt
 5: Exit

Enter 1, 2, 3, 4, or 5: 1

Enter deposit amount: 0
ERR: Minimum deposit is $100.00 !!

Enter deposit amount: 111.11

Seth's Account#:123 has a balance of $123.11

********** CAMPUS CASH CARD **********
 1: Deposit
 2: Dine
 3: Kayak
 4: Email receipt
 5: Exit

Enter 1, 2, 3, 4, or 5: 4
Email of transaction sent to seth@ucsd.edu

********** CAMPUS CASH CARD **********
 1: Deposit
 2: Dine
 3: Kayak
 4: Email receipt
 5: Exit

Enter 1, 2, 3, 4, or 5: 5
Thanks for using Campus Cash!
```

Submit the final version of your program as p8.c

Verify you SAVED your work in the Documents - cs5f HOME directory.