

Assignment 4

COP 3514 Program Design

Due 06/21/2013

Blue font represents a variable

Red font represents a constant

Green font represents a function

1. In this assignment you are given the task to complete a simple messenger program. This assignment will evaluate your knowledge in program design and structures. You are provided with 3 files: User.h User.c and main.c . You are to modify only User.c . User.h defines 3 structures and 10 functions. You are to complete the function definitions in User.c . There are a total of 10 users and each of the users is stored in an array called `users[]` defined in User.h. The `id` variable in User represents the index of that user inside the `users[]` array.
 - The `verify_date()` function returns 1 if the date is correct, otherwise it returns 0. The `year` must be between 1900 and 2013, the `month` has to be between 1 and 12. The `day` depends on the month and should be validated accordingly (for February assume 28 days).
 - The `verify_name()` function converts the name to lower cases and returns 1 only if there are only letters in the name (`first_name` and `last_name`). If there are any other characters besides letters the function returns 0.
 - The `verify_username()` function returns 1 only if there are only letters and numbers in the `username`. If there are any other characters besides letters and/or numbers the function returns 0.

- **verify_user_data()** function returns 1 only if all of the above verification functions return 1. The **verify_user_data()** also prints "**ERROR: Date of birth**" and/or "**ERROR: First name**" and/or "**ERROR: Last name**" and/or "**ERROR: Username**" in case there are verification errors.
- The **create_date()** function creates a Date structure.
- The **create_msg()** function creates a Message structure.
- The **create_user()** function creates a **user**. If there are verification errors then the valid variable in the user structure is set to 0, otherwise it is set to 1.
- The **show_user_info()** function prints the user info. The format should be:

```

\t\t\tUser info:
\t\t\tDate of birth: [month]/[day]/[year]
\t\t\tHeight: [height]
\t\t\tWeight: [weight]
\t\t\tFirst Name: [first_name]
\t\t\tLast Name: [last_name]
\t\t\tUsername: [username]
\t\t\tEmail: [email]

```

The “\t\t\t” means that there needs to be 3 tab spaces in each line. [month], [day] and [year] are part of the Date structure of the User. [height], [weight], [first_name], [last_name], [username], [email] are part of the User structure.

- The **send_message()** function sends a message from user with user id **_sender** to user with user id **_receiver** by changing the **message** variable in the user with user id **_receiver**. The **Message** structure has two variables. The **from** variable represents the **username** of the user with user id **_sender**, and the **content** variable is the content of the message.

- The **read_message()** function prints the message content from the user with user id **_user_id**. The format should be:
- ```

\t\t\t[content]

```

The “\t\t\t” means that there needs to be 3 tab spaces in the line and [content] is the message content of the user with user id **\_user\_id**

### EXTRA CREDIT (10 points):

-The **verify\_email()** function returns 1 if the email format is valid. A valid email format should be username@mail.usf.edu where username can contain only letters and numbers.

Files:

**You are provided with the files:**

```

User.c
User.h
main.c
Makefile.

```

**You are to complete the functions in User.c**

**You are NOT to modify User.h**

**You can use main.c to test the correctness of the functions.**

## **Grading:**

**Your code will be tested with 9 regular test case inputs (10 points each) and 2 special test case inputs (5 points each)**

## **Submission:**

**Submission should be done only through Blackboard. Submit only User.c**

## **Compiling and testing:**

**To compile your file make sure all of the files are in the same directory. Type “make --always-make” in the terminal in the directory of the files. If there are no compilation errors a main executable will be created.**

**To test your program run “./main” from the terminal in the directory of the files.**