

# Discussion V - C File Organization

---

Version 2.0

September 29, 2014

## 1 THE LOCK

Create a project where a user is presented with a welcome screen asking for an initial password. The password will consist of only four letters **U,D,L,R** but is limited to 25 such characters.

Once the user sets the initial password, the following scenarios should occur in sequence

1. Encrypt the password and store the password in a variable
2. Clear the screen and prompt the user for a password
3. Decrypt the stored password and compare with the input in step 2
4. If the passwords match, present the user with a "**secret message**"
5. If they don't match keep prompting the user to enter the right password
6. When the wrong password is entered a third time, present the user with a failure message

## 2 FORMATTING YOUR PROJECT FILES

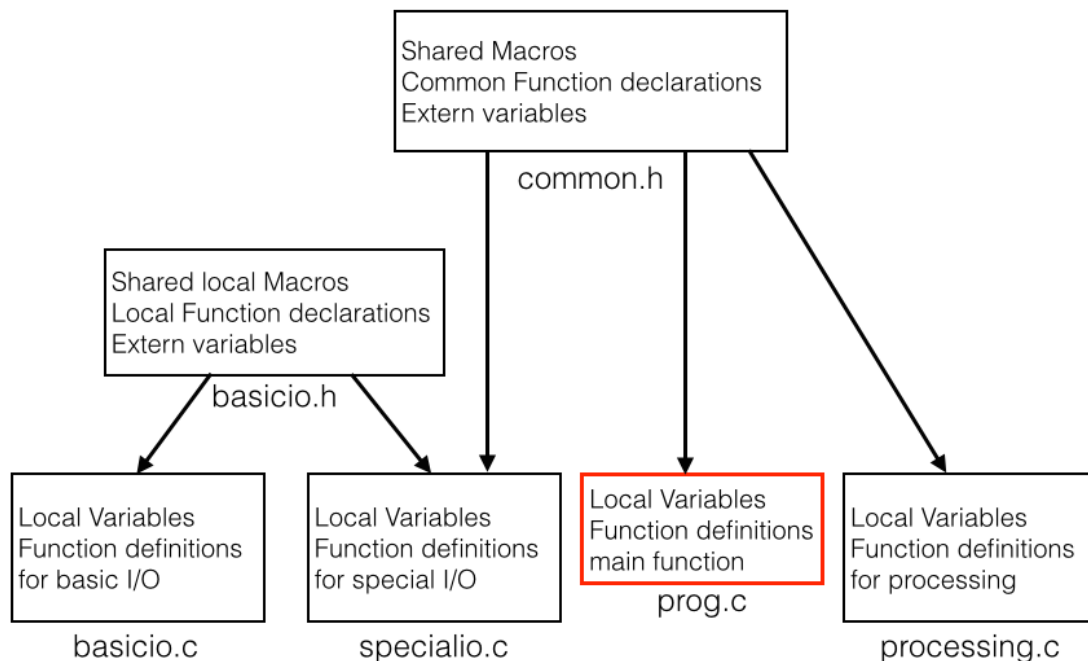
The format for any header file used in your project must include the following

```
#ifndef _HEADERNAME_H (The definition macro)
#define _HEADERNAME_H
Shared variables, function declarations etc. (use extern where-ever applicable)
#endif
```

## 3 ORGANIZING YOUR PROJECT FILE CONTENTS

There is no hard and fast rule for project file organization. However, content could be appropriately distributed across files based on functionality. This would ease the process of debugging long sequences of code, facilitating code reuse.

Following is a possible organization of code in multiple project files



Arrows depict header file inclusion in associated c files.

**common.h** includes a select list of functions that need to be shared from **specialio.c** or **prog.c** or **processing.c**, among each other.

## 4 COMPILING YOUR PROJECT FILES

Use the following command in the console to compile your project files

```
gcc -Wall -o projname main.c proj1.c proj2.c
```