

Functions in C – continued

As discussed in the previous note, functions are to be used to help simplify our development of code and to aid in the portability of code between programs. Different programs have need for more or less from functions, and over the next several classes we will be seeing how to advance from simple functions to more complex ones which can do more within our programs. The terms being used to describe the operations of functions here are not formal, but are meant to aid in the understanding of how to create your own functions.

Level One Functions

At the very simplest level, functions can perform simple tasks which are not dependent on data within your code, but require multiple lines of output. A simple example of this would be a function which outputs a simple introduction to a program.

```
#include <stdio.h>

//Function Prototypes
void intro (void);           //A simple message displaying function

int main (void)
{
    intro();                 //A call of the function

    system("PAUSE");        //The program then returns here
    return 0;
}

/*****
Function:    intro
Does:       Displays a simple introduction to the program
Receives:   Nothing
Uses:       Nothing
Returns:    Nothing
*****/
void intro (void)
{
    printf("Welcome to the Function Introduction Program.");
    printf("This program will teach the basics of function use");
    printf("Press return to continue...");
    getchar();
}
```

Try to follow the style convention as shown above for both layout and information about the functions you create. It will make it MUCH simpler when collaborating with others on programs.

Assignment

Using appropriate conventions, write C programs which:

1. For a fictitious game, displays an `intro()` message when starting up, displays the `rules()` of the game after the user presses enter, and then displays the `controls()` for playing after another press of the enter key.
2. Allows a user to see 5 menu options to select, from 1 to 5, and will display the message “You chose Option x”, where x is a number from 1 to 4. If 5 is entered, it will display a message and exit the program. Any other value entered should generate an error message. Use the functions `showMenu()`, `choice1()`, `choice2()`, `choice3()`, `choice(4)` as a minimum.
3. Adapts the program from question 2, adapts the choices to the following:
 - `choice1()` generates a random number between a user entered low and high
 - `choice2()` lets the user enter a word and the function will print it out in reverse
 - `choice3()` takes a user entered string and tests if it could be a number (ex “154 is good, 1d9 is bad”)
 - `choice4()` lets the user enter three integers and will display them from smallest to largest.