EE 2372 Software Design I
Assignment #4: Functions Programs
Due Date: Monday October 30 at 11:59 PM

Objectives:

1) Learn write functions that include arguments
2) Learn to use pointers as arguments for functions

Description

This programming assignment will have you write two programs. These two programs are problems 3 and 6 of Chapter 9 with some additions:

1. Write a function that takes three arguments: a character and two integers. The character is to be printed. The first integer specifies the number of times that the character is to be printed on a line, and the second integer specifies the number of lines that are to be printed. Write a program that makes use of this function. Call the program charblock.c
2. Write and test a function that takes the addresses of three double variables as arguments and that moves the value of the smallest variable into the first variable, the middle value to the second variable, and the largest value into the third variable. Write a program that reads in 3 double values, and then prints the values in sorted order using the function you just created. Call the program dsort.c

Deliverables:

1) Submit the C source code as an attachment to our TA Pavithra Pochamreddy at ppochamredd@miners.utep.edu. Name your programs charblock.c and dsort.c.
2) The subject line must be "EE 2372 Assignment 4" (don't include the quotes).

Scoring:

Your grade for this assignment will be determined by three criteria. The first criterion determines if your program compiles and runs producing the correct result. The correct result must adhere to what is specified in the **Tasks** section. The second criterion is whether the program follows the interface specification outlined in the **Tasks** section. The third criterion determines if your source code is well documented. Your source code must include (at the top) your name, class section, due date, assigned date, and a small description of your program. For this assignment, each line of code should have a descriptive comment.

Operation/Successful Demonstration	**60%**
Was a C source file submitted? 10%	
Does the program compile on Linux? 15%	
Does the program run correctly on Linux? 35%	

Adherence to Interface Specification	**30%**
Does the program adhere to the interface specification in the tasks section? 30%	
Comments	**10%**
Is the source code well-documented? 10%	
Lateness	**10%** per day (including weekends and holidays)