

Lab Seven

Pointers to Functions

CS2263, SUMMER 2020

Learning Outcomes

At the conclusion of the lab, students should be able to

- Write functions to be used as arguments to functions
- Pass pointers to functions function arguments
- Use functions passed as arguments.

Setting Up

In Lab5 you formalized your understanding of strings into a module containing the typedef for a `String` and the package of functions needed to manage Strings. One of these was `compareStrings()` which you then used to sort the strings using the `qsort()` function. Make sure you have access to the `String` module. Also make sure that you have access to your `Point2D` module.

Exercise One

Create a standalone function (not associated with a module) that sorts using your favourite sorting algorithm (*I know that you have one!*). Test it using a stack-declared array of integers in a simple test program.

```
$ sortTest
```

Submit:

- A screen shot of the make command output for a successful compile
- A screen shot of a successful program run

Exercise Two

Modify your sorting function so that, like `qsort()`, you can pass a pointer to a comparison function as a parameter. You will need to do some online research to discover the technique to do this. Searching for *C pointers to functions* should do the trick. Using your program from Lab5 Exercise 4 (`stringListSortTest`), call your sorting function instead.

Submit:

- A screen shot of the make command output for a successful compile
- A screen shot of a successful program run

Exercise Three

Based on your programs from Lab6, create a program based on your Point2D module that creates an array of random Point2D values, passes the array to your sorting function, along with a comparison function as a parameter. Note that you'll need to write the comparison function for the Point2D data type. For our purposes here, simply compare the x-values of the coordinate.

```
$ sortPoint2D 50
```

Submit:

- A screen shot of the make command output for a successful compile
- A screen shot of the program's successful run for 50 values.

Submission

Before 8:30am on the day after this lab, students should submit to the LMS a zip/tar file (named LastName_FirstName_Lab7.zip or LastName_FirstName_Lab7.tar) containing

- the required material for each question (use the headings indicating the question number) in a single pdf file (named LastName_FirstName_Lab7.pdf)
- the source code for the programs and makefile.
- Please do not include the binary, object, or executable files in the submission