# CS 253: Introduction to Systems Programming
# cch – A Utility for Classifying Characters

Jim Conrad

September 5, 2019

## Learning Objectives

- The GNU C toolchain
- The `main` function in C
- Automatic variables, string literals, and the `int` data type
- Expressions
- I/O functions `getchar` and `printf`
- Control statements including if-else, loops, switch-case

## Overview

In this project, you will develop a new Linux utility, `cch`, for classifying characters read from stdin and writing the results to stdout. Here is an example of how to execute `cch` to classify the characters in, testFile.txt:

```
jconradmbp13:solution kq7b$ cch <testFile.txt
upper-case:  26
lower-case:  26
vowels:  12
consonants:  40
digits:  10
total:  67
jconradmbp13:solution kq7b$
```

The cch utility has no arguments. Each character read from stdin can be classified using a `switch` (or `if`) statement and counted. The output reports the number of characters in each classification and the total number of characters read. The example above finds 26 upper-case characters, 26 lower-case letters, 12 vowels, 40 consonants, 10 digits, and a total of 67 characters were read from stdin.

## Getting Started

Create a directory called `p1` for your project. In that directory, download the skeletal `cch.c` starter file and `testFile.txt` from the blackboard assignment. Implement your solution by adding code to the supplied source file. If your code reads from stdin using `getchar` then you won't need to use arrays in this project. C does not support exceptions a la java; K&R Chapter 1 explains how to handle `EOF`.

You will need bash I/O redirections to arrange for cch to read from a disk file. E.g.:

```
cch <testData.txt
```

## Submission

Submit your completed project using the `submit` tool on onyx:

1. `cd p1` if you have not already done so on onyx
2. Remove any files you don't wish to submit
3. `submit username cs253 p1` to submit the contents of your current directory to an instructor whose onyx account is, username.
4. You can verify a submitted assignment with, `submit -check username cs253 p1`

Note: The onyx `man submit` command will display help for the submit tool. Artifacts that must be submitted include:

- cch.c

Note: Onyx submissions are checked for plagiarism. Plagiarized projects receive 0 points and might be referred to the Dean of Students Office. Don't allow your friends, roommate, etc. to copy your code. However, it is perfectly acceptable to post screenshots of code snippets of non-working code on piazza where you may request assistance. It is not acceptable to publicly post this or any entire project on a publicly accessible web site (e.g. github, facebook, whatever). I realize this can be confusing, and different instructors have different policies, so ask when questions arise.

## Grading

- Your source file must be named exactly, cch.c
- Your source file compiles without warning or error messages
- Do not submit binary files (object or executable)
- It's a good practice to submit your test data files
- Your solution will be tested with the instructor's test data files

- One test data file, testFile.txt, will be provided on blackboard
- Your solution must label the various classifications (e.g. "upper-case:") exactly as noted in the overview above (else the grader's test program may deduct for missing features)
- Test your solution carefully to ensure it classifies characters correctly
- Your solution should return a 0 (normal) exit status to the shell if the program executes correctly
- Your solution should handle the `EOF` condition

## References

The lecture slides illustrate how to program `main` to return a 0 (normal) status. Refer to K&R Chapter 1 when you get stuck.

There is help available on onyx for the `getchar` API using `man getchar`. You will likely need to study the API's documentation to determine when `getchar` reaches the End-of-File (EOF).

K&R is your best resource for printf.