March 18, 2018  
CS4513 – Hugh C. Lauer

Project 1: Dumpster DIving

# Design

To get started right away, just tap any placeholder text (such as this) and start typing.

## Programs / Scripts (pseudo-code)

### Error Calls: checks for following constraints, if something goes wrong, ERROR message and prints usage message for each utility tool.

#### ERROR\_call(): default to usage message and exits

#### ERROR\_no\_file(int argc): checks if file or directory present

#### ERROR\_rename\_call(): checks if rename call fails

#### ERROR\_remove\_call(): checks if remove call fails

#### ERROR\_open\_call(): checks if open call fails

#### ERROR\_opendir\_call(): checks if opendir call fails

#### ERROR\_mkdir(): checks if mkdir call fails

#### ERROR\_duplicate\_dir(): checks if directory already present

#### ERROR\_rmdir\_call(): checks if rmdir call fails

#### ERROR\_limit\_dumpster(): limit dumpster size to 10 same file or directory

#### ERROR\_stat\_call(): checks if stats are passed when moving

#### ERROR\_chmod\_call(): checks if permissions are passed when moving

#### ERROR\_utime\_call(): checks if utime call fails

#### ERROR\_unlink\_call(): checks if unlink call fails

#### ERROR\_fileNotFound(char\* source\_path): checks if file is in dumpster

#### ERROR\_rw\_file(): checks if can read or write to file

#### ERROR\_fopen\_call(): checks if open file

### Rm

#### usage(): prints usage message(with -h or error) “./rm -f, -h, -r < file(s) >

#### flag\_check(int arc, char\*\* argv): check flag input

#### check\_f\_flag(char\* file): check if -f flag is present

#### check\_r\_flag(char\* file): check if -r flag is present

#### set\_dumpster(): set the dumpster to be used to place files or directories

#### check\_dumpster(): check to see where the dumpster is located

#### set\_file(char\* file): set the file to be moved

#### get\_extension(char\* path): get extension is duplicates

#### get\_dumpster\_path(char\* file, char\* dumpster\_path, char\*\* new\_path): dumpster path

#### copyto\_dump(char\* file, char\* dumpster\_path, struct stat file\_stat): move to dumpster

#### remove\_directory(char\* current\_path, char\* current\_dumpster\_path, int same): remove dir(s)

#### remove\_force(char\* directory): remove files or directories permanently

### Dv

#### usage(): prints usage message(with -h or error) “./dv -h < file(s) >

#### remove\_directory(char\* current\_path, char\* file, int same): remove dir(s)

### Dump

### To easily apply any text formatting you see in this outline with just a tap, on the Home tab of the ribbon, check out Styles.

### For example, this paragraph uses Heading 3 style.

## Number of Runs

## Data Recording Method

## System Conditions

# Results

To get started right away, just tap any placeholder text (such as this) and start typing.

## Tables or Graphs

## Time (milliseconds) to Make, Remove and Recover files and directories.

## Calculations (Mean and STD)

# Analysis

To get started right away, just tap any placeholder text (such as this) and start typing.

## Describe what the results mean

## How does your prediction match measured? If there is a difference, why do you think there is?