EXPLANATION OF BASH SCRIPT

## VALIDATE DIRECTORY LOCATION

The first section of the script invokes the Select\_Input\_File a function that prompts the user to enter the location of the rectangle.txt file. The script checks that what has been entered is a valid directory and if not, prompts the user to re-enter the directory location.

sed is then used to remove excess forward slashes from the user input for the directory location and appends rectangle.txt to form the file location. The command that is used is:

Sed -i -E “s|/\*$|/rectangle.txt|g” dirpath.txt

* The purpose of this command is to replace excess forward slashes from the $directory\_path variable to ensure that it is a valid
* The -i tag enabled the sed function to make permanent changes to text in dirpath.txt.
* The -E tag enables the sed command to using regular expressions (regex)
* The ‘|’ is used as an alternative separator to make the statement easier to read.
* The command itself is a substitute command that searches for all lines that match the regex expression /\*$ which matches the presence of one of more ‘/’ at the end of the line.
* Any matches are then replaced with the string ‘/rectangle.txt’
* The ‘g’ means that **all** matches will be replaced in the file.

## REMOVE HEADERS FROM RECTANGLE.TXT

The next significant action is the next sed command which strips the first line of the rectangle.txt which contains the headers for the data. The command that is used is:

Sed -i ‘1d’ “$directory\_path”

* $directory\_path variable (validated directory entered by user).
* The ‘1d’ value deletes the first line in the $directory\_path (which represents the location of the rectangle.txt) this line contains the headers.

## PROCESSING THE DATA

Using an UNTIL loop (with the Internal Field Separator value set as “,”) data is read from the rectangle.txt and saved to rectangle\_temp.txt. in a more suitable format.

The data is manipulated using the sed\_processing function which contains the following code:

echo $1 >> $temp\_rec

sed -iE "s|^$1|$2: $1$3|1" $temp\_rec

The first line stores the value from the 1st argument in the function to a temporary file.

The next line is the sed function which formats the data ready for output.

sed -iE “s|^$1|$2: $1$3|1” $temp\_rec

* The -iE flags allow the changes to be saved to the file and the use of regex.
* The substitution searches for a line that begins with the $1 variable and then replaces it with the string $2 value followed by a colon and then the value of the $1 variable followed by $3 an end of line character.
* The 1 at the end means that the changes will only be changed for the first instance of a pattern match.

For example, in the main body of the script the first call to the sed\_processing uses the arguments: $name “Name” and “” Which results in the following sed command being executed:

sed -iE “s|^Rec1|Name: Rec1|1” rectangle\_temp.txt

## DISPLAYING THE FORMATTED DATA

To display the formatted data AWK was used. The BEGIN section sets up the title for the output and the END section is used for formatting for presentation.

The main body of the awk command is:

{printf "%s %5s | %s %2d | %s %d | %s %5d | %s %s\n", $1,$2,$3,$4,$5,$6,$7,$8,$9,$10}

It uses the printf function to align the variables ($1 - $10) in a presentable manner using ‘|’ as a column divider.