# Comp 150

# Assignment 3

# Harmanjit Singh

# December 03, 2023

Assignment 3

Problem solving:

* mean = sum (input List from main function)/Total length.
* Median for even number:

middle1 = income List [length of the list // 2 - 1]

middle2 = income List [ the list // 2

median value = (middle1 + middle2) / 2

* Median for odd number:

Median value = income List [length of the list // 2]

* variance = sum ((x – mean value) \*\* 2 for x in input List) / Total length of the list
* Standard deviation = sqrt(variance)

Design:

* Start time to get calculate the processing time of function
* Call the main function
* Assigned the directory path of main ‘xls’ file in file name
* Assigned the extension name of the file.
* Call the generateJSON function from skeletom.py to generate Json file from the ‘xls’ file.
* Take file name, root directory and extension from the main file.
* Generate sheet and header by calling each function separately.
* Compose the file
* Try to open the file in written mode if not open, print file not found! And quit the whole function
* Try to write the file and print JSON file successfully generated. If not, Print JSON file is not successfully generated.
* Call the extragrossincomeList function from skeletom.py to get gross income list.
* Try to open the JSON file in read mode and take all gross income of the employee one by one and append it in the one list called ExtragrossincomeList.
* Call the computeIncomeList function from taxUtilities.py to get net income list of employees.
* Get gross income from the JSON file and subtract all deductions from it one by one and store it in one list.

Deductions:

1. Provincial tax:

* Apply if statement and calculate provincial tax.
* Return provincial tax.

1. Federal tax:

* Apply if statement and calculate federal tax.
* Return federal tax.

1. CPP:

* CPP = 0.0595\*gross Income.
* CPP = min (3754.45, CPP) to get minimum CPP.
* Return CPP.

1. EI:

* EI = 0.0163\*gross Income.
* EI = min (1002.45, EI) to get minimum EI.
* EI = round (EI,2) to round the value to 2 decimal places.
* Return EI

1. Health premium:

* Apply if statement and calculate Health premium.
* Return health premium.
* Call the mean function from nstatistics.py file to calculate mean of the gross income list.
* Get the length of the list.
* If length = 0, mean = 0.
* Else mean = sum (input List from main function)/Total length.
* Return mean
* To calculate standard deviation of the gross income list, call the standard\_deviation function from nstatistics.py file.
* To get median of the gross income list, call the median function from the nstatistics.py file.
* Sorted the gross income list.
* Get the length of the list.
* If length of the list %2 = 0 (means even), take two incomes from the middle of the list and sum them. Then divide them to get median income.

middle1 = income List [length of the list // 2 - 1]

middle2 = income List [ the list // 2

median value = (middle1 + middle2) / 2

* + - * If the length is in odd number, simply take the exact middle number of the list and assign it in median variable.

median value = income List [length of the list // 2]

* + - * Return median.
* To get mode of the gross income list, call the computeModes function from the nstatistics.py file.
* Count occurrence of each number.
* Find the maximum frequency.
* Find the numbers with maximum frequency and assign that value in modes.
* Return modes
* Display all the average Gross Incomes, average net income, average all modes for the employees, population variance for employees. Median for employees.
* Call the GradeAssignment() function from qualityAssurance.py to get assessment of the whole file.
* Print the whole-time duration that takes to run the program.

Flow charts that are new in assignment 3:







