# Lab 5 Pseudocode

## Van Robbins 9/30/2024

### Executable File

import lab5Utility; def mainMenu(first): if first is true it is the first time the program is being run: print(“Walcome to Van’s Employee Databasecan create or look up entries in the database”) findOrCreate=input(“Would you like to| Create| Look Up| About Data”); checkInput= lab5Utility.check(findOrCreate,1,3); ensures Input is valid, 1 or 2 else first is not true and the program has been run before, adds option to exit program findOrCreate=input(“Would You Like To| Create| Look Up| About Data| Exit”); checkInput= lab5Utility.check(findOrCreate,1,4); ensures entry is valid between 1 and 3

if checkInput==True: If checkinput is found to be valid convert to int and send to function based on input  
 validInput=int(findOrCreate);  
 if validInput==1:  
 create();#Go to create   
 elif validInput==2:  
 look();  
 elif validInput==3:  
 about();  
 else:  
 exit();  
else: Input is invalid the check function will ask the user if they would like to try again  
 if checkInput is a special value send back to retry  
 mainMenu(False);  
 else:  
 quit();

Create Employee Profile def create(): Open text file for editing print(“’s create this employees profile!”) FIRSTNAME firstName=input(“What is the employee’s first name?”); Checks that Input is 35 Characters or less with utility file function checkInputLength checkFirstNm= lab5Utility.checkInputLength(firstName,35); while firstName is not valid: if checkFirstNm is a special value retry firstName=input(“What is the employee’s first name?”); checkFirstNm= lab5Utility.checkInputLength(firstName,35); else: quit(); LASTNAME lastName=input(“What is the employee’s last name?”); Checks that Input is 35 Characters or less with utility file function checkInputLength checklastNm=lab5Utility.checkInputLength(lastName,35); while lastName is not valid: if checkFirstNm is a special value retry lastName=input(“What is the employee’s last name?”); checklastNm=lab5Utility.checkInputLength(lastName,35); else: quit(); AGE age=input(f”How old is {firstName} {lastName}?“); Checks that age is between 16 and 120 with utility function check checkAge= lab5Utility.check(age,16,120); while checkAge is not valid: if checkAge is a special value retry age=input(f”How old is {firstName} {lastName}?“); checkAge= lab5Utility.check(age,16,120); else: quit(); TITLE title=input(f”What is {firstName} {lastName}’s title?“); Checks that Input is 35 Characters or less with utility file function checkInputLength checktitle=lab5Utility.checkInputLength(title,35); while checktitle is not valid: if checkAge is a special value retry title=input(f”What is {firstName} {lastName}’s title?“); checktitle=lab5Utility.checkInputLength(title,35); else: quit(); SALARY salary=input(f”What is {firstName} {lastName}’s yearly salary“); Checks that Input is greater than 0 with checkPositiveNonZero checkSalary= lab5Utility.checkPositiveNonZero(salary); while checkSalary!=True: if checkSalary==”goAgain”: salary=input(f”What is {firstName} {lastName}’s yearly salary“); checkSalary= lab5Utility.checkPositiveNonZero(salary); else: quit(); Add employee info to end of text file in this order |||

|;

print("Employee Successfully Added");  
Tell the user how many employees are in the database  
  
mainMenu(False);

Looks for employee in database def look(): dataSearch = open(“dataLab5.txt”,“r+”); Opens data text file in read only mode data=dataSearch.read();Sets current database to data string

print("\nLet's look for this employee!")  
firstName=input("First Name:\n");  
lastName=input("Last Name:\n");  
  
Search using upper version of data and input  
firstLastName=(firstName+"|"+lastName).upper();  
upperData=data.upper()  
  
See if employee in data  
if(firstLastName in upperData):  
 print("Employee Found");  
 startIndex=upperData.find(firstLastName); Finds Index where first and last name is in uppercase data  
 Cuts data into parts given the regular data, the start of the data wanted, the end character, and divider character, returns employees data as array employeeInfo, each part of employee info in different array index  
  
Employee not in data  
else:  
 print("\nEmployee Not Found\n")  
dataSearch.close(); Closes text file  
  
SENDS USER BACK TO MAIN MENU FUNCTION False becauase not first time running  
mainMenu(False);

Tells user about data def about(): Opens text file in mode to read data See how many entries are in the database currentNumberEntries=lab5Utility.amountEntry(dataSearch.read(),“;”); dataSearch.close(); find average salary with findAverageSalary function print(f”About the Data:’s {currentNumberEntries} employees in the system!“); print(f”The average salary is ${averageSalary}“) mainMenu(False);

def findAverageSalary(entries): dataSearch = open(“dataLab5.txt”,“r+”);#Opens text file in mode to read data allData=dataSearch.read(); index=0; totalSum=0 Find last entry of all seperate emloyees info add to sum totalSum=totalSum+personSalary; average=totalSum/entries; dataSearch.close();#Closes text file return average

RUNS Initial main menu function, True since first time running mainMenu(True);

### Utility File

def check(sizeInput,validInputBottom,validInputTop): Check input is between the top and bottom numbers current=validInputBottom; validSize=[]; add all numbers between top and bottom numbers to validSize array if sizeInput not in validSize: Checks input is one of the numbers in array, if not print(f”The input must be between {validInputBottom} and {validInputTop}“); again=input(”Would you like to try again Y/N“); if(again==”Y”): return “goAgain”; else: return False; else: Number is valid return True;

def checkPositiveNonZero(enteredValue): Check that value can be converted into a number If can, do so num=float(enteredValue); else Tell person to try again if num is less than 0 print(“The input must be greater than 0”); again=input(“Would you like to try again Y/N”); if(again.upper()==“Y”): return “goAgain”; else: return False; Number is greater than 0 return True;

def checkInputLength(enteredString, maxLength): Check that string is less than the maxLength alotted if len(enteredString)>maxLength: print(f”The input can not be longer than {maxLength} characters”); again=input(“Would you like to try again Y/N”); if(again==“Y”): return “goAgain”; else: return False; else: return True;

def amountEntry(data, divider): index = 0; count = 0; search through all data and count all the dividers while (index < len(data)): if (data[index]) == divider: count+=1; index +=1; return count;

Cuts string data based on the start and end’s index, divides whole into parts based on divider def cut(data,start,end,divider): dataCutToStart=data[start:];#Cuts Data from start index parsedData=[];Empty array that will store parsed data index=0; start=0; if type(end)==int:end is already defined as an index value end=end; else:End is not a defined index but a character endEntryCharacter=end; end=dataCutToStart.find(endEntryCharacter);Find first occurence of character from already the start of the data dataCut=dataCutToStart[start:end+1]; Cuts data from start to found end+1 to include end character Loop looks through cut data dividing data into parts based on given divider while index!=len(dataCut): if(dataCut[index])==divider:If the character at the index of the data is the same as the divider append that part to array parsedData.append(dataCut[start:index]); start=index+1; For next data part the start will be at the end of the last part index+=1; elif((dataCut[index])==endEntryCharacter) or (index==end):Saves last part of data parsedData.append(dataCut[start:end]); index+=1; else: Character is neither divider or end index+=1; return parsedData; Returns each part of data in array entry