Jun Zhou

Illinois Institute of Technology

ITMD513

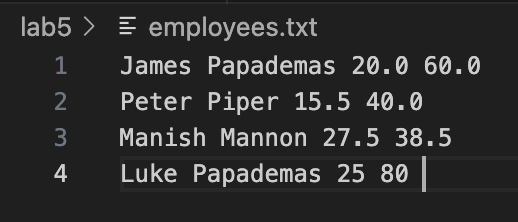
Email: zhoujun9633@163.com

CWID: A20488371

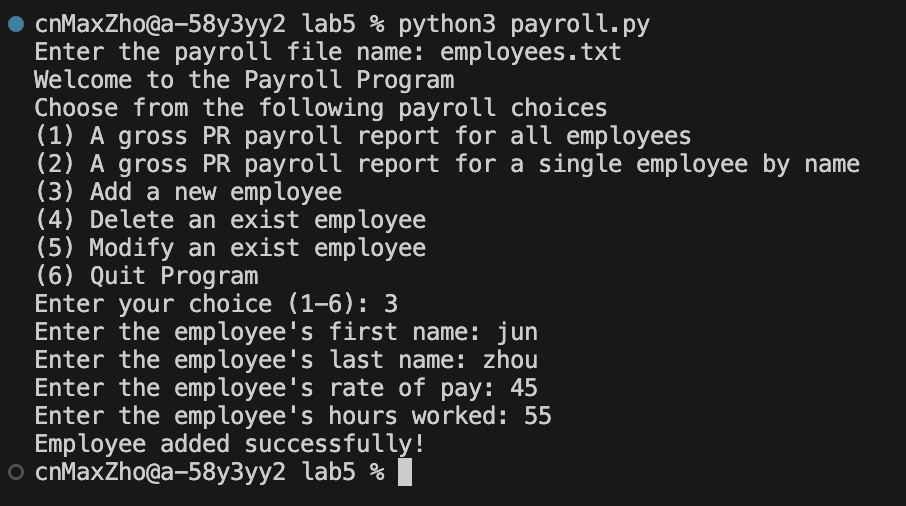
# Snapshot

1. Run option 3 and add in yourself as a employee when prompted. Include your name appropriately, rate of pay and hours worked and have the data added to your employees.txt file.

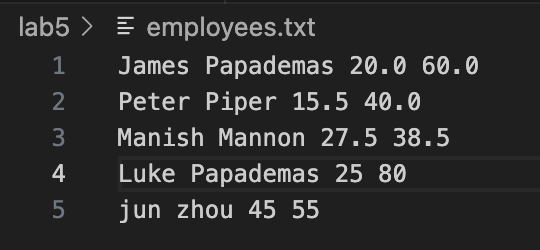
employees.txt file before add emp:



output after add emp:

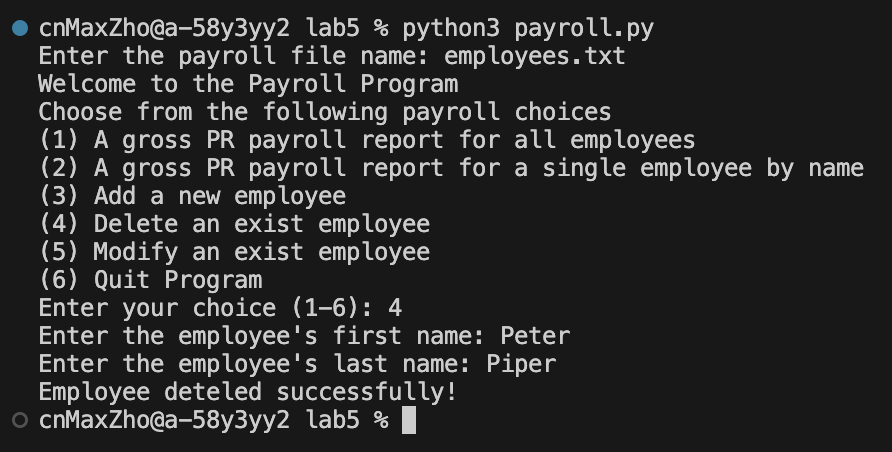


employees.txt file after operation:

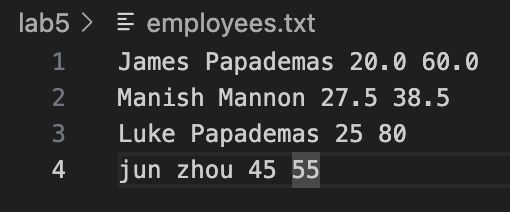


1. Run option 4 and delete Peter Piper from the employees.txt file.

output after delete Peter Piper:

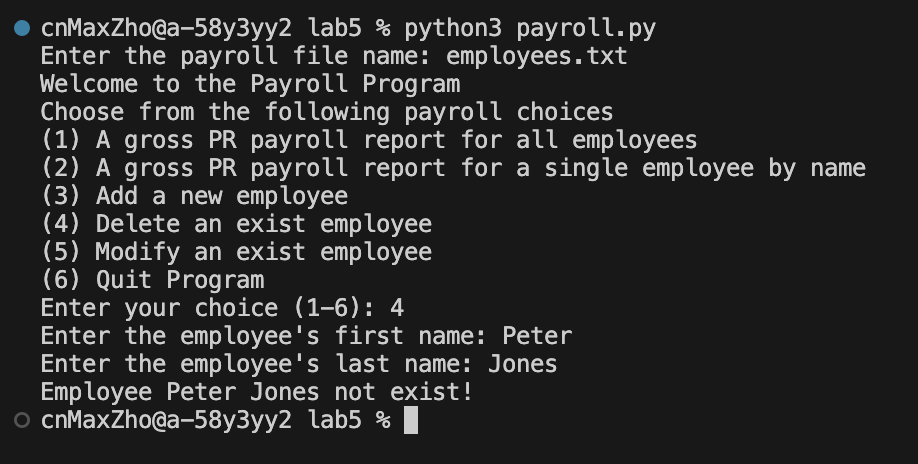


employees.txt file after operation:

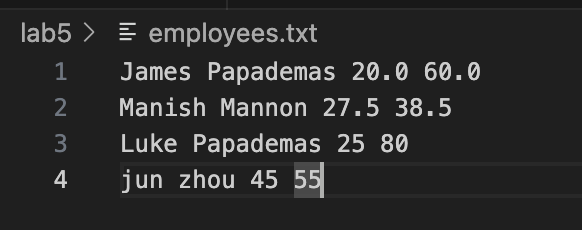


1. Run option 4 again and attempt to delete Peter Jones, a former employee from the file. There should be an error message to the screen to show the employee does not exist.

output after try delete Peter Jones:

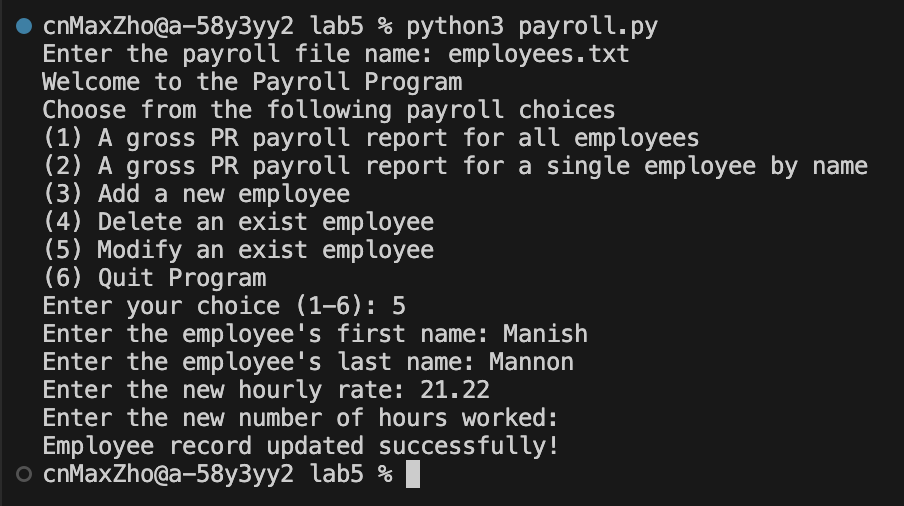


employees.txt file after operation:

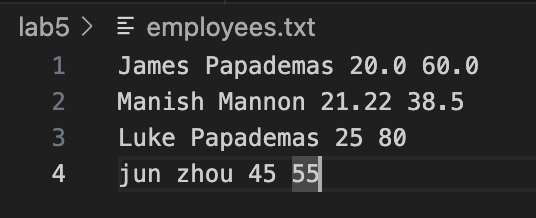


1. Run option 5 modify Manish Mannon and change his pay rate to 21.22.

output after modify Manish Mannon:

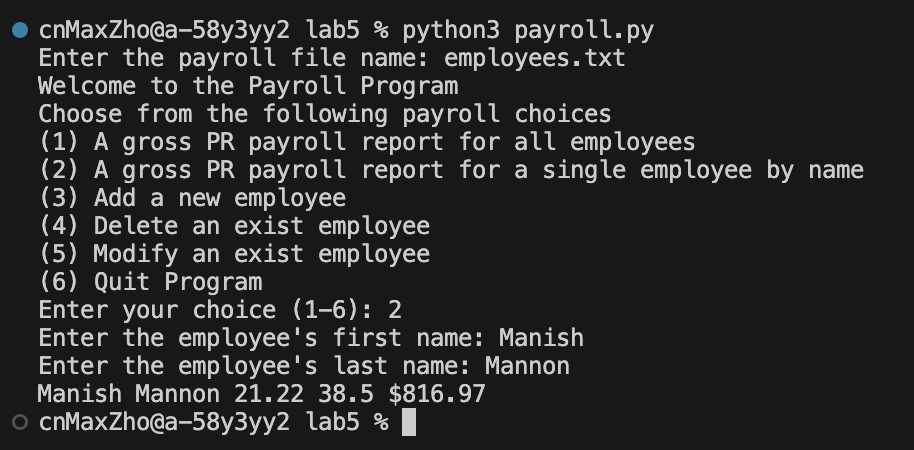


employees.txt file after operation:

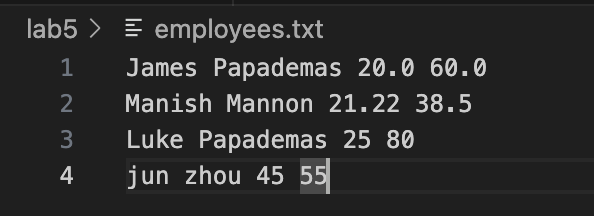


1. Run option 2 and show the employee result for Manish Mannon.

output of showing record of Manish Mannon:

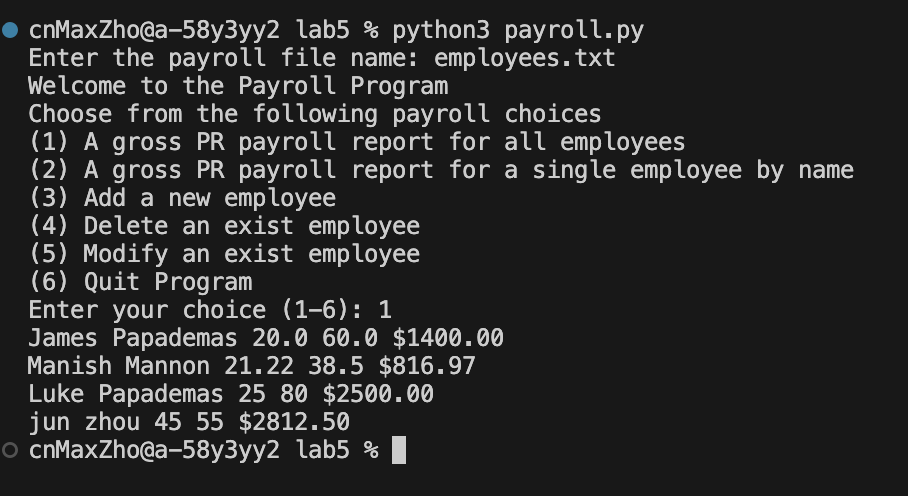


employees.txt file after operation:

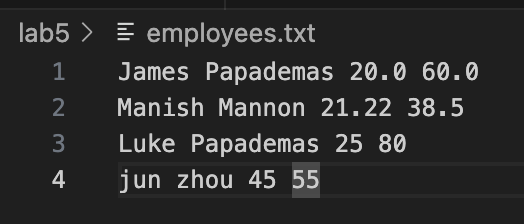


1. Run option 1 and show all employee results.

output of showing all recores:



employees.txt file after operation:



# Sourcecode

payroll\_functions.py

import os

# Jun Zhou

# June 26th, 2023

# Lab5

# ITMD513

def calculate\_gross\_pay(rate=None, hours=None):

"""

Calculates the gross pay for an employee based on their hourly rate and hours worked.

If rate or hours are not provided, prompts the user to enter them.

Args:

rate (float): The hourly rate for the employee.

hours (float): The number of hours worked by the employee.

Returns:

float: The calculated gross pay.

"""

if rate is None:

rate = float(input("Enter the hourly rate: "))

if hours is None:

hours = float(input("Enter the number of hours worked: "))

gross\_pay = 0

if hours > 40:

overtime\_hours = hours - 40

regular\_pay = rate \* 40

overtime\_pay = (rate \* 1.5) \* overtime\_hours

gross\_pay = regular\_pay + overtime\_pay

else:

gross\_pay = rate \* hours

return gross\_pay

def print\_all(file\_name=None):

"""

Prints the payroll information for all employees in a given file.

If file\_name is not provided, prompts the user to enter it.

Args:

file\_name (str): The name of the payroll file.

Raises:

FileNotFoundError: If the payroll file is not found.

ValueError: If the payroll file has an invalid format.

"""

if file\_name is None:

file\_name = input("Enter the payroll file name: ")

try:

emp\_file = open(file\_name, "r")

for line in emp\_file :

emp\_info = line.split(" ")

grossPay = calculate\_gross\_pay(float(emp\_info[2]), float(emp\_info[3]))

print("%s %s %s %s $%4.2f" % (emp\_info[0], emp\_info[1], emp\_info[2], emp\_info[3].strip(), grossPay))

emp\_file.close()

except FileNotFoundError:

print("Error: Payroll file not found!")

except ValueError:

print("Error: Invalid payroll file format!")

def print\_emp(file\_name=None):

"""

Prints the payroll information for a specific employee in a given file.

If file\_name is not provided, prompts the user to enter it.

Args:

file\_name (str): The name of the payroll file.

Raises:

FileNotFoundError: If the payroll file is not found.

ValueError: If the payroll file has an invalid format.

"""

if file\_name is None:

file\_name = input("Enter the payroll file name: ")

try:

emp\_file = open(file\_name, "r")

inputFirstName = input("Enter the employee's first name: ")

inputLastName = input("Enter the employee's last name: ")

line = emp\_file.readline()

for line in emp\_file :

emp\_info = line.split(" ")

if (inputFirstName == emp\_info[0] and inputLastName == emp\_info[1]):

grossPay = calculate\_gross\_pay(float(emp\_info[2]), float(emp\_info[3]))

print("%s %s %s %s $%4.2f" % (emp\_info[0], emp\_info[1], emp\_info[2], emp\_info[3].strip(), grossPay))

break

emp\_file.close()

except FileNotFoundError:

print("Error: Payroll file not found!")

except ValueError:

print("Error: Invalid payroll file format!")

def add\_emp(file\_name=None):

"""

Adds a new employee to the payroll file.

If file\_name is not provided, prompts the user to enter it.

Args:

file\_name (str): The name of the payroll file.

Raises:

FileNotFoundError: If the payroll file is not found.

ValueError: If the payroll file has an invalid format.

"""

if file\_name is None:

file\_name = input("Enter the payroll file name: ")

try:

emp\_file = open(file\_name, "r")

temp\_file = open("temp.txt", "a")

inputFirstName = input("Enter the employee's first name: ")

if inputFirstName == "":

inputFirstName = "default"

inputLastName = input("Enter the employee's last name: ")

if inputLastName == "":

inputLastName = "default"

inputRate = input("Enter the employee's rate of pay: ")

inputHours = input("Enter the employee's hours worked: ")

employee\_exists = False

for line in emp\_file:

emp\_info = line.split(" ")

if (inputFirstName == emp\_info[0] and inputLastName == emp\_info[1]):

employee\_exists = True

break

if employee\_exists:

print("Employee already exists in the payroll!")

return

else:

emp\_file.seek(0)

for line in emp\_file:

temp\_file.write(line)

temp\_file.write("\n")

temp\_file.write("%s %s %s %s" % (inputFirstName, inputLastName, inputRate, inputHours))

print("Employee added successfully!")

emp\_file.close()

temp\_file.close()

os.remove(file\_name)

os.rename("temp.txt", file\_name)

except FileNotFoundError:

print("Error: Payroll file not found!")

except ValueError:

print("Error: Invalid payroll file format!")

def delete\_emp(file\_name=None):

"""

Deletes an employee from the payroll file.

If file\_name is not provided, prompts the user to enter it.

Args:

file\_name (str): The name of the payroll file.

Raises:

FileNotFoundError: If the payroll file is not found.

ValueError: If the payroll file has an invalid format.

"""

if file\_name is None:

file\_name = input("Enter the payroll file name: ")

try:

emp\_file = open(file\_name, "r")

temp\_file = open("temp.txt", "w")

inputFirstName = input("Enter the employee's first name: ")

inputLastName = input("Enter the employee's last name: ")

employee\_exists = False

for line in emp\_file :

emp\_info = line.split(" ")

if (inputFirstName == emp\_info[0] and inputLastName == emp\_info[1]):

employee\_exists = True

continue

else:

temp\_file.write(line)

emp\_file.close()

temp\_file.close()

os.remove(file\_name)

os.rename("temp.txt", file\_name)

if employee\_exists:

print("Employee deteled successfully!")

else:

print("Employee %s %s not exist!" % (inputFirstName, inputLastName))

except FileNotFoundError:

print("Error: Payroll file not found!")

except ValueError:

print("Error: Invalid payroll file format!")

def modify\_emp(file\_name=None):

"""

Modifies the payroll information for an employee in the payroll file.

If file\_name is not provided, prompts the user to enter it.

Args:

file\_name (str): The name of the payroll file.

Raises:

FileNotFoundError: If the payroll file is not found.

ValueError: If the payroll file has an invalid format.

"""

if file\_name is None:

file\_name = input("Enter the payroll file name: ")

try:

emp\_file = open(file\_name, "r")

temp\_file = open("temp.txt", "w")

inputFirstName = input("Enter the employee's first name: ")

inputLastName = input("Enter the employee's last name: ")

found = False

emp\_file.seek(0)

for line in emp\_file:

emp\_info = line.split(" ")

if (inputFirstName == emp\_info[0] and inputLastName == emp\_info[1]):

found = True

newRate = input("Enter the new hourly rate: ")

if newRate == "":

newRate = emp\_info[2]

newHours = input("Enter the new number of hours worked: ")

if newHours == "":

newHours = emp\_info[3]

temp\_file.write("%s %s %s %s\n" % (inputFirstName, inputLastName, newRate, newHours))

else:

temp\_file.write(line)

if found:

print("Employee record updated successfully!")

else:

print("Employee %s %s not found in the employees." % (inputFirstName, inputLastName))

emp\_file.close()

temp\_file.close()

os.remove(file\_name)

os.rename("temp.txt", file\_name)

except FileNotFoundError:

print("Error: Payroll file not found!")

except ValueError:

print("Error: Invalid payroll file format!")

payroll.py

import os

import payroll\_functions

# Jun Zhou

# June 26th, 2023

# Lab5

# ITMD513

def menu():

"""

Displays the payroll menu options.

"""

pstr = "Welcome to the Payroll Program\n"

pstr += "Choose from the following payroll choices\n"

pstr += "(1) A gross PR payroll report for all employees\n"

pstr += "(2) A gross PR payroll report for a single employee by name\n"

pstr += "(3) Add a new employee\n"

pstr += "(4) Delete an exist employee\n"

pstr += "(5) Modify an exist employee\n"

pstr += "(6) Quit Program"

print (pstr)

def process\_option(choice, file\_name):

"""

Processes the user's choice and calls the corresponding payroll function.

Parameters:

choice (str): The user's choice.

file\_name (str): The name of the payroll file.

"""

if (choice == "1") :

payroll\_functions.print\_all(file\_name)

elif (choice == "2") :

payroll\_functions.print\_emp(file\_name)

elif (choice == "3") :

payroll\_functions.add\_emp(file\_name)

elif (choice == "4") :

payroll\_functions.delete\_emp(file\_name)

elif (choice == "5") :

payroll\_functions.modify\_emp(file\_name)

elif (choice == "6") :

os.\_exit(0)

else :

print("Invalid Choice")

os.\_exit(0)

def main():

file\_name = input("Enter the payroll file name: ")

menu()

option = input("Enter your choice (1-6): ")

process\_option(option, file\_name)

if \_\_name\_\_ == "\_\_main\_\_":

main()