Lab 3

PART A – In-Class (Due End of Class), 5 marks

1. Create a Python Project with the name Lab3
2. Create a Python file called lab3.py
3. In file lab3.py Create function main in the format:

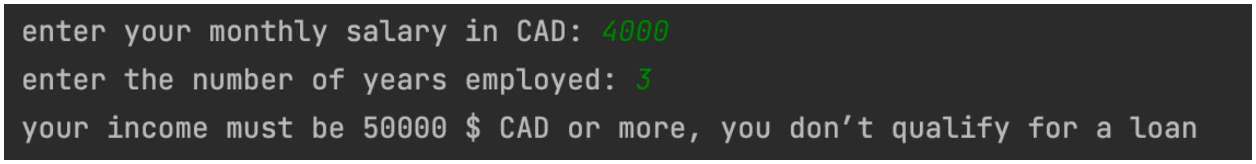
```
def main():
    pass


if  __name__ == “__main__”:
    main()
```

Create the following functions:

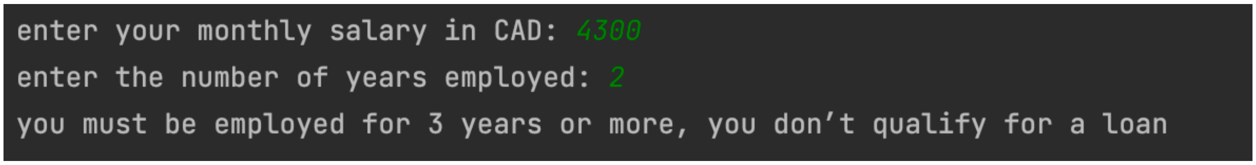
- Function qualifies_for_loan(), the function asks the user for two pieces of data: (1) the monthly salary in CAD, and (2) the number of years employed. The function will calculate the annual income of the employee (note: use a CONSTANT for the number of months in a year):
    - If the employee’s annual income is 50000 $ CAD or more and he/she has been employed for 3 years or more (use CONSTANTs for 50000.0 and 3) then the function will display the message “you qualify for a loan “ and return from the function
    - If the annual income is less than 50000$ CAD, the message “your income must be 50000 $ CAD or more, you don’t qualify for a loan” will be displayed and return from the function
    - If the employee has been employed for less than 3 years, the message “you must be employed for 3 years or more, you don’t qualify for a loan” will be displayed and return from the function

Note: in order to compare user input to numbers, change the user input to a float or int like this: month_salary_cad = int(input(“Monthly salary cad?”))

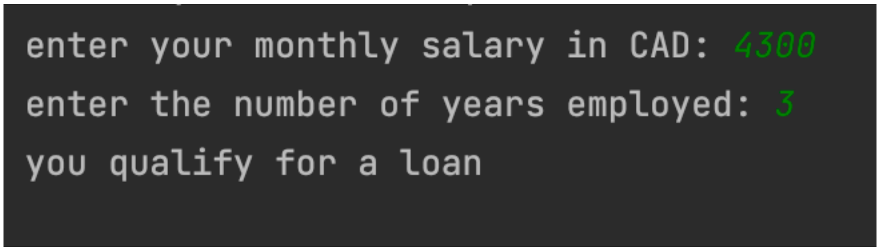
Sample output of function qualifies_for_loan()



```
enter your monthly salary in CAD: 4000
enter the number of years employed: 3
your income must be 50000 $ CAD or more, you don’t qualify for a loan
```



```
enter your monthly salary in CAD: 4300
enter the number of years employed: 2
you must be employed for 3 years or more, you don’t qualify for a loan
```

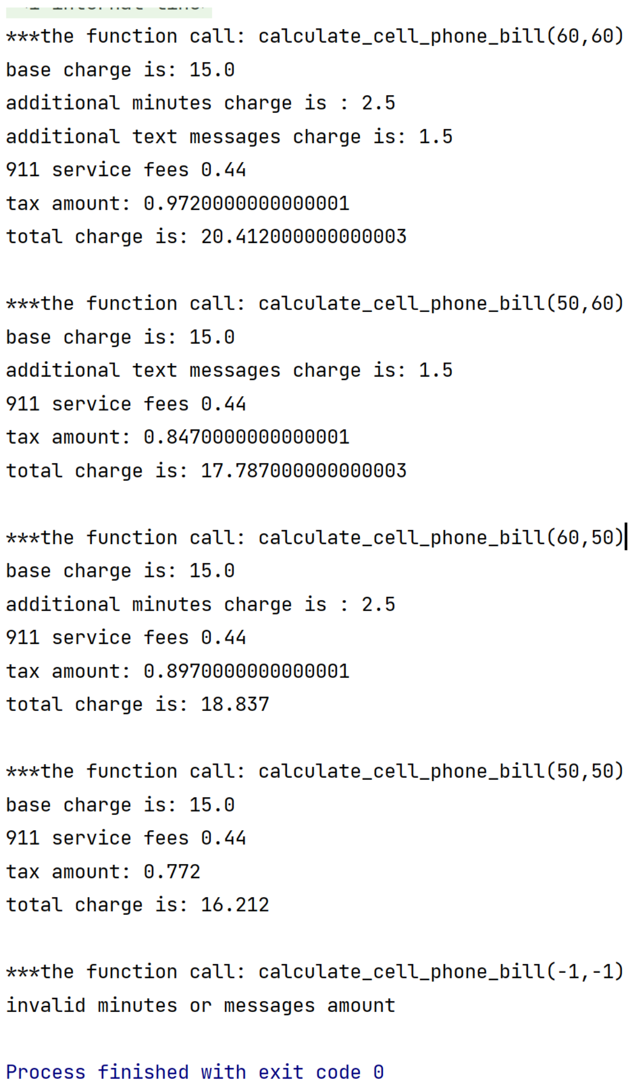


```
enter your monthly salary in CAD: 4300
enter the number of years employed: 3
you qualify for a loan
```

- Function calculate_montly_cell_phone_bill_charge_cad(), the function takes two parameters , the *air time minutes amount* and the *text messages amount*.
  - If either of those values was negative, the function displays the message (“invalid minutes or messages amount”) and the function ends.
  - The cellphone plan includes 50 minutes of airtime and 50 text messages for $15.0 a month
  - Each additional minute of airtime costs $0.25, while each additional text message costs $0.15
  - The cellphone bills include an additional charge of $0.44 to support the 911 call centers
  - The entire bill (including the 911 charge) is subject to 5% (0.05) sales tax

The function will calculate the monthly bill and display the base charge, additional minutes charge (if any), additional text messages charge (if any), the 911 fee, the tax and the total bill charge.
Only display the additional minute and text message charges if the user incurred costs of these categories

Sample output of the function:



```
***the function call: calculate_cell_phone_bill(60,60)
base charge is: 15.0
additional minutes charge is : 2.5
additional text messages charge is: 1.5
911 service fees 0.44
tax amount: 0.9720000000000001
total charge is: 20.412000000000003

***the function call: calculate_cell_phone_bill(50,60)
base charge is: 15.0
additional text messages charge is: 1.5
911 service fees 0.44
tax amount: 0.8470000000000001
total charge is: 17.787000000000003

***the function call: calculate_cell_phone_bill(60,50)
base charge is: 15.0
additional minutes charge is : 2.5
911 service fees 0.44
tax amount: 0.8970000000000001
total charge is: 18.837

***the function call: calculate_cell_phone_bill(50,50)
base charge is: 15.0
911 service fees 0.44
tax amount: 0.772
total charge is: 16.212

***the function call: calculate_cell_phone_bill(-1,-1)
invalid minutes or messages amount

Process finished with exit code 0
```

Note: You may assume that the given parameters are valid, and you don’t need to validate them

- In the function main(), remove the keyword pass and call functions loan_qualifier and calculate_montly_cell_phone_bill_charge_cad().

Make sure to include doc-string in every function

Demo to your instructor the output from the two functions on today before you leave.

Include the code with your Part B submission.

Part B – Take Home (before next lecture), 10 marks

Add the following function to the file lab3.py

- Function play_chicago(), the function implements the Chicago game. The game is played for 11 rounds. Round numbers start from 2 to 12. In each round two random numbers between 1 and 6 inclusive will be generated, the random numbers and the round number will be displayed to the user. If the sum of the two numbers equals the round number, the user wins a point and a message to that effect will be displayed. If the user does not win a point a message stating that will also be displayed. At the end of each round the user will be asked if he/she wants to play another round. If the user response was no or the round number is 12 the game will end and the total number of earned points will be displayed.

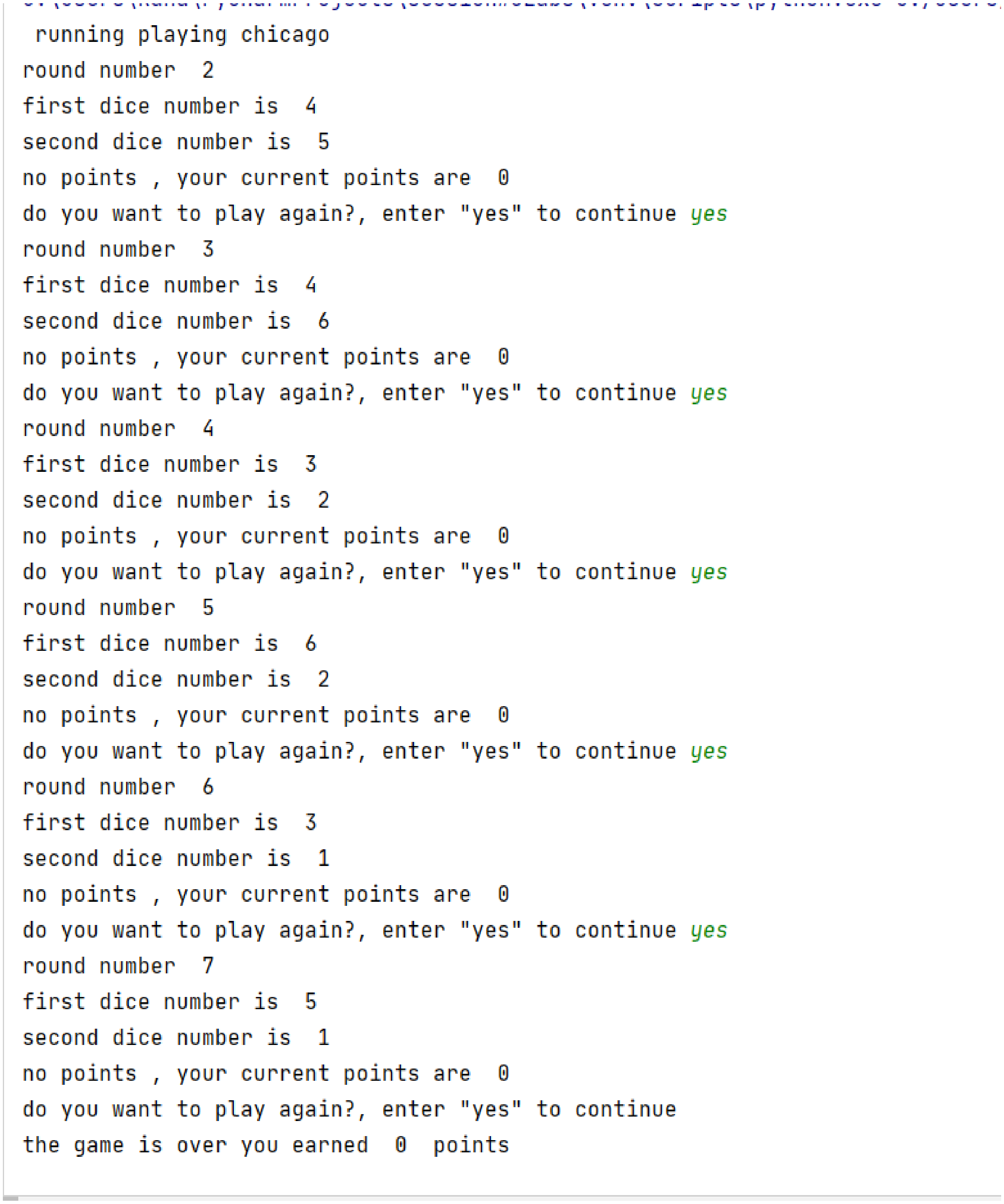
  NOTE: You must use a **while loop** to handle the 11 rounds (from 2 to 12).

  NOTE: to generate a random number of between 1 and 6 inclusive:

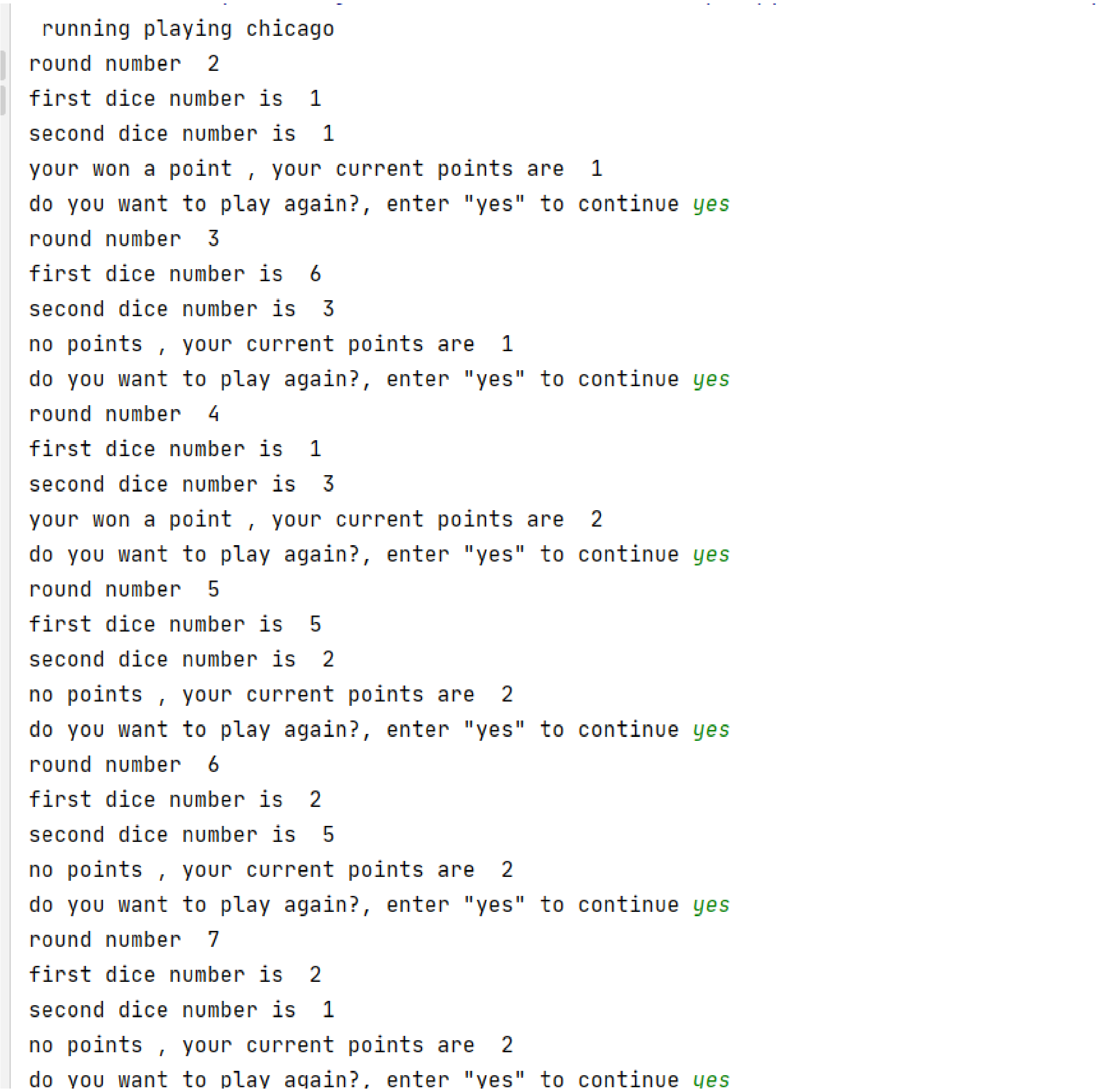
  - Import module random at the top of the file
  - Use the statement

    number = random.randint(1,6)

Demo: Sample output of play_chicago:



```
 running playing chicago
round number  2
first dice number is  4
second dice number is  5
no points , your current points are  0
do you want to play again?, enter "yes" to continue yes
round number  3
first dice number is  4
second dice number is  6
no points , your current points are  0
do you want to play again?, enter "yes" to continue yes
round number  4
first dice number is  3
second dice number is  2
no points , your current points are  0
do you want to play again?, enter "yes" to continue yes
round number  5
first dice number is  6
second dice number is  2
no points , your current points are  0
do you want to play again?, enter "yes" to continue yes
round number  6
first dice number is  3
second dice number is  1
no points , your current points are  0
do you want to play again?, enter "yes" to continue yes
round number  7
first dice number is  5
second dice number is  1
no points , your current points are  0
do you want to play again?, enter "yes" to continue
the game is over you earned  0  points
```



```
 running playing chicago
round number  2
first dice number is  1
second dice number is  1
your won a point , your current points are  1
do you want to play again?, enter "yes" to continue yes
round number  3
first dice number is  6
second dice number is  3
no points , your current points are  1
do you want to play again?, enter "yes" to continue yes
round number  4
first dice number is  1
second dice number is  3
your won a point , your current points are  2
do you want to play again?, enter "yes" to continue yes
round number  5
first dice number is  5
second dice number is  2
no points , your current points are  2
do you want to play again?, enter "yes" to continue yes
round number  6
first dice number is  2
second dice number is  5
no points , your current points are  2
do you want to play again?, enter "yes" to continue yes
round number  7
first dice number is  2
second dice number is  1
no points , your current points are  2
do you want to play again?, enter "yes" to continue yes
```

Add the function call play_chicago to the main function.

Best Practices

- Variable names are descriptive and should be lower_snake_case
- function names should be lower_snake_case
- All functions include Doc-String comments

Part B Submission

Submit your lab3.py file to the Lab3 Dropbox on the Learning Hub before next lecture.