HIGHER NATIONALS - ASSESSMENT (ASSIGNMENT)

Name of the Learner	Mr. / Ms. Mfm.Amsath		
Ref. No. of the Learner	Ait221514	**Pearson Regd. No.**	
Unit Title	**Unit 01: Programming**	**Batch No & Semester**	**CSD 15**
Assignment Number	01 (All LOs)	**Assessor**	Mr. A.R.M. Nizzad
Issue Date	26.10.2022		
Submission Date	31.11.2022	**Date Received 1st submission**	:
Re-submission Date		**Date Received 2nd submission**	:

Assessor Summative Feedback:			
Grade:	**Assessor Signature:**	**Date:**	
Resubmission Feedback - Formative:			
Grade:	**Assessor Signature:**	**Date**	
Internal Verifier's Comments:			
Signature of the IV		**Date**	
Student Agreement: I understand the feedback given to me and agree to carry out the actions in future works as required and indicated.		**Student Signature** **Date**	

Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

LEARNER ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each learner must sign a declaration confirming that the work is their own.

Please list the evidence submitted for each task. Indicate the page numbers where the evidence can be found or describe the nature of the evidence (e.g. video, illustration).

Unit No. & Title:							
Task	Assessment Evidence	**P**	*Page No.*	**M**	*Page No.*	**D**	*Page No.*

Additional comments to the Assessor:

PLAGIARISM

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Learner declaration

I certify that the work submitted for this assignment is my own. I have clearly referenced any sources used in the work. I understand that false declaration is a form of malpractice.

Learner Signature: Date:

ASSESSMENT TRACKING

ASSESSMENT RECORD AND FEEDBACK SHEET					
Programme:	HND Computing	**Student Name:**			
Unit No. & Title:	01. Programming	**Assessment Date:**		**Unit Grade:**	
Assessor Name:	Mr. A.R.M. Nizzad	**Completion Date:**		**IV Signature:**	

Assignment	Learning Objectives	Criteria Targeted	Date Issued	Hand In Date	Formative Feedback	Resubmission Date*
* Resubmissions must be approved by the Assessment Board						

Assignment No.	Grading Criteria	Date Acheived	Comments	Assessor Signature

Unit Review Plan				ISVU 2014 Rev 1.0
Assignment	Task	Evidence	Formative Comments	Date

Task 1

1. based on the scenario provided.

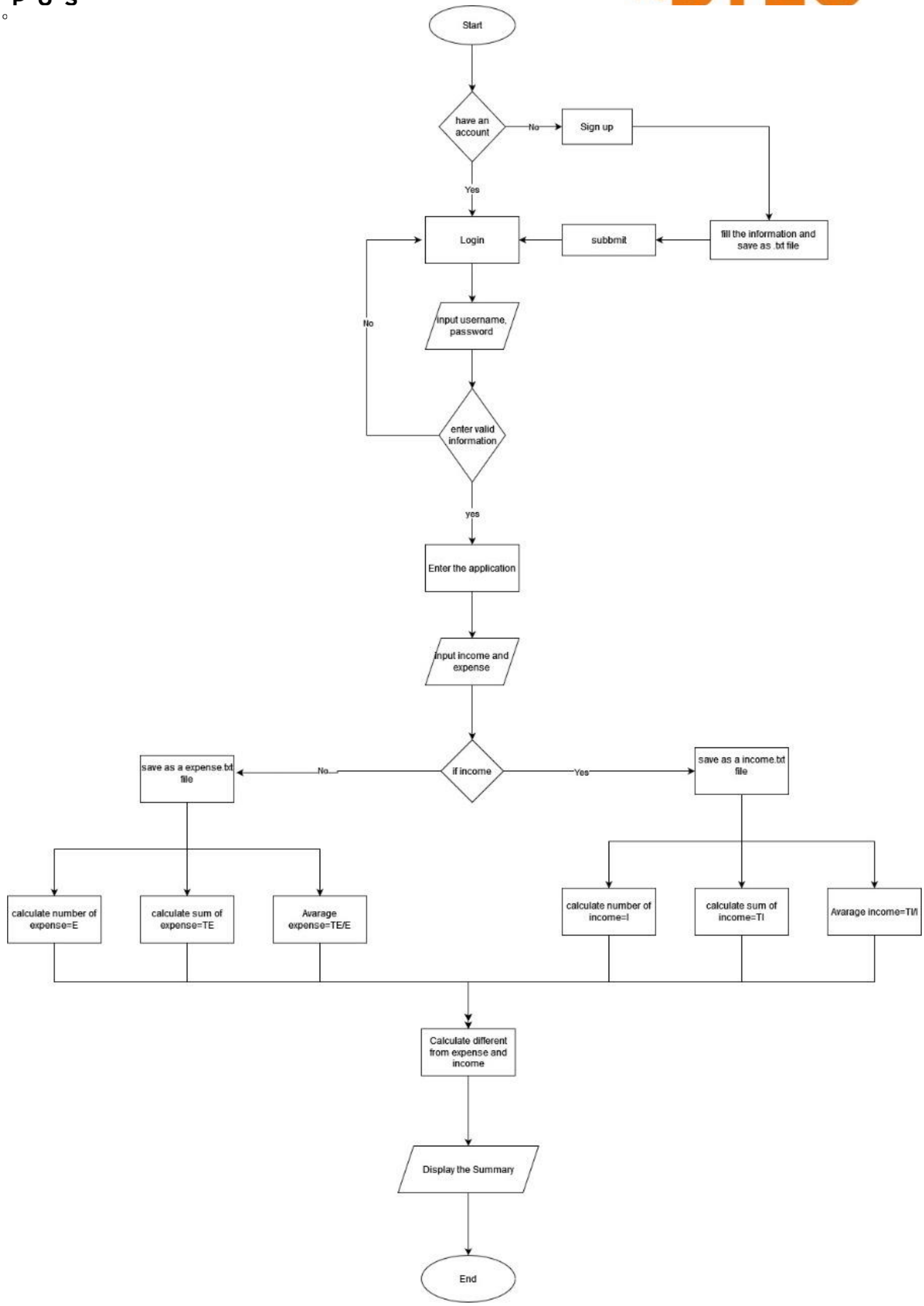
an algorithm is

A sequence of steps used to solve a problem is called an algorithm. In this case, I'm using a "for" loop to retrieve the number of lines in the "summary.py" file as well as inputs from text files line by line. After that, the system will deliver the anticipated outcomes using simple mathematics. An algorithm is a mathematical procedure used in this circumstance to solve a problem.

II. The procedure for creating an application

1. Recognized the query
2. gathered specifications
3. Requirements analysis 4. Algorithm development
5. Flowcharted the system representation
6. Python was used to create the system.

3.write the algorithm to reflect the above scenario using pseudo code or flow chart


Start
have an account
No
Sign up
Yes
Login
subbmit
fill the information and save as .txt file
input username, password
No
enter valid information
yes
Enter the application
input income and expense
save as a expense.txt file
No
if income
Yes
save as a income.txt file
calculate number of expense=E
calculate sum of expense=TE
Avarage expense=TE/E
calculate number of income=I
calculate sum of income=TI
Avarage income=TI/I
Calculate different from expense and income
Display the Summary
End

Analyze the relationship between the written algorithm and the code variation as well as the implementation of the algorithm in a suitable language.
When referring to code in computer science, the term "algorithm" is typically used to describe a highly optimized design, such as a widely utilized solution to a challenging problem. The phrase describes software that adheres to stringent design guidelines and solves challenging issues.

Code is typically repetitive and basic. Codes, for instance, are frequently easy to apply. Algorithms are more complex and may start as a research project or another important undertaking. An algorithm is typically not considered to be any impromptu code produced by a developer that does not solve a significant problem.

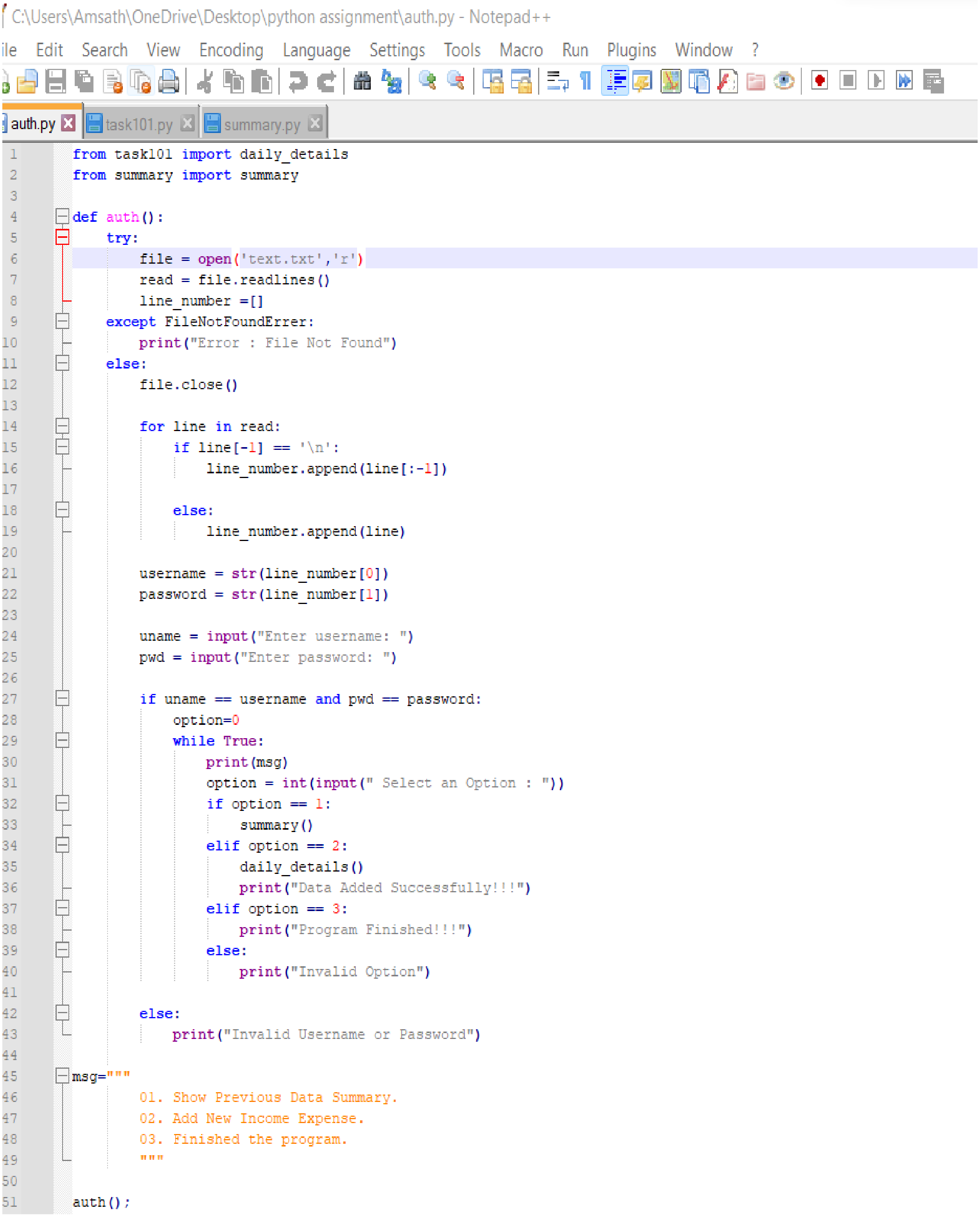
Task 2

2.1 Describe the procedural, object-oriented, and event-driven paradigms, as well as their similarities and differences.

In procedural programming, a program is defined as a collection of phases, including its subprograms. Conversely, declarative programs aim to describe the output without regard for the procedures required to computerize it, instead providing some description or denotation of the desired outcome.
A technique for organizing code based on encapsulation, inheritance, replacement, programming to interfaces, and other concepts is known as object-oriented programming. Most procedural programming is used in object-oriented programming.
Event-based programming requires creating event handling procedures and utilizing the core event loop of the underlying system. By utilizing numerous libraries, you can avoid the challenge of creating your own event loop. They currently work with the event loop given by the system. Even if it is not always the case, object-oriented programming is regularly used to create event-based applications.

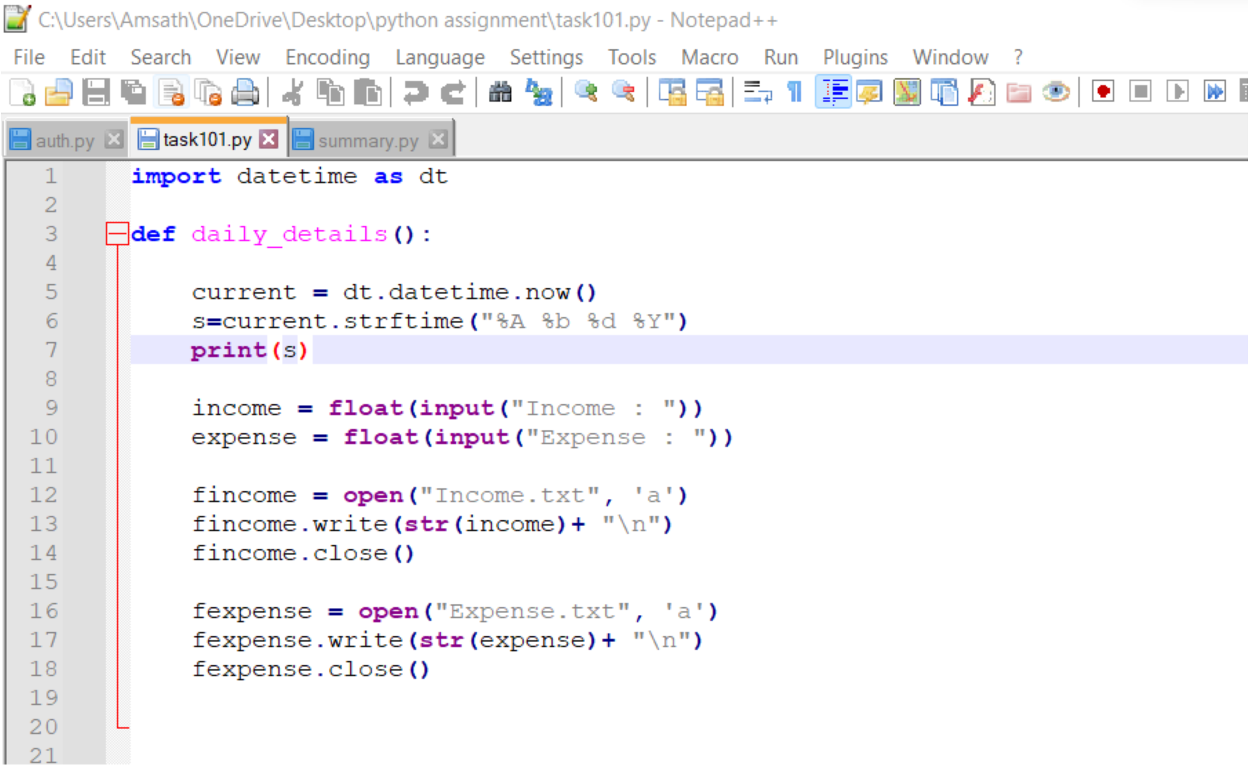
2.2 Based on your Task 1 algorithms, compare and contrast the procedural, object-oriented, and event-driven paradigms.
Small functions make up the program, and the code is reusable. Data abstraction is a notion used by this system. Simple to add features. The system is very nearly object-oriented.

auth.py


C:\Users\Amsath\OneDrive\Desktop\python assignment\auth.py - Notepad++
ile Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
auth.py
task101.py
summary.py

```python
from task101 import daily_details
from summary import summary

def auth():
    try:
        file = open('text.txt','r')
        read = file.readlines()
        line_number =[]
    except FileNotFoundErrer:
        print("Error : File Not Found")
    else:
        file.close()

        for line in read:
            if line[-1] == '\n':
                line_number.append(line[:-1])

            else:
                line_number.append(line)

        username = str(line_number[0])
        password = str(line_number[1])

        uname = input("Enter username: ")
        pwd = input("Enter password: ")

        if uname == username and pwd == password:
            option=0
            while True:
                print(msg)
                option = int(input(" Select an Option : "))
                if option == 1:
                    summary()
                elif option == 2:
                    daily_details()
                    print("Data Added Successfully!!!")
                elif option == 3:
                    print("Program Finished!!!")
                else:
                    print("Invalid Option")

        else:
            print("Invalid Username or Password")

msg="""
        01. Show Previous Data Summary.
        02. Add New Income Expense.
        03. Finished the program.
        """

auth();
```

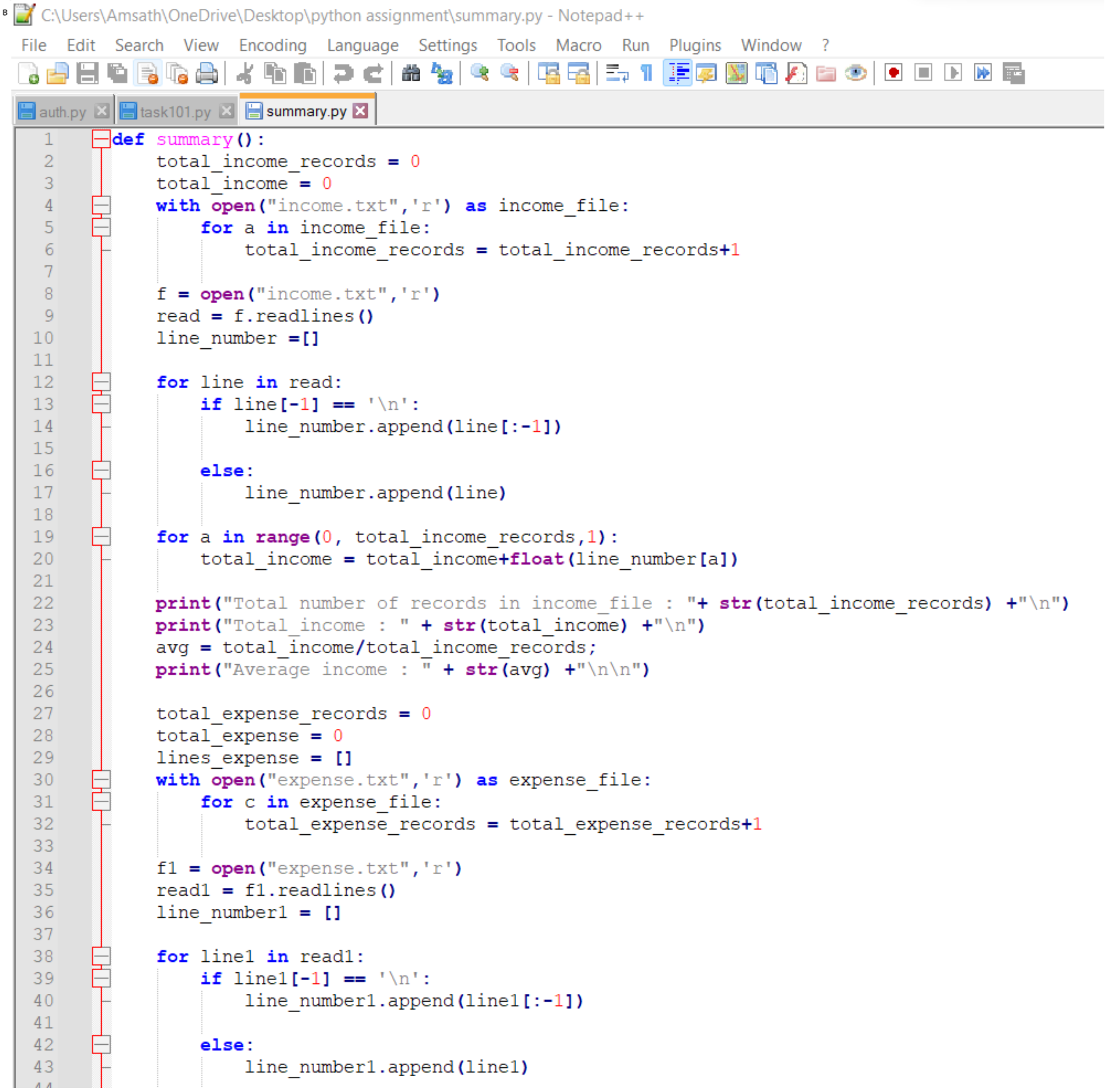
task101.py



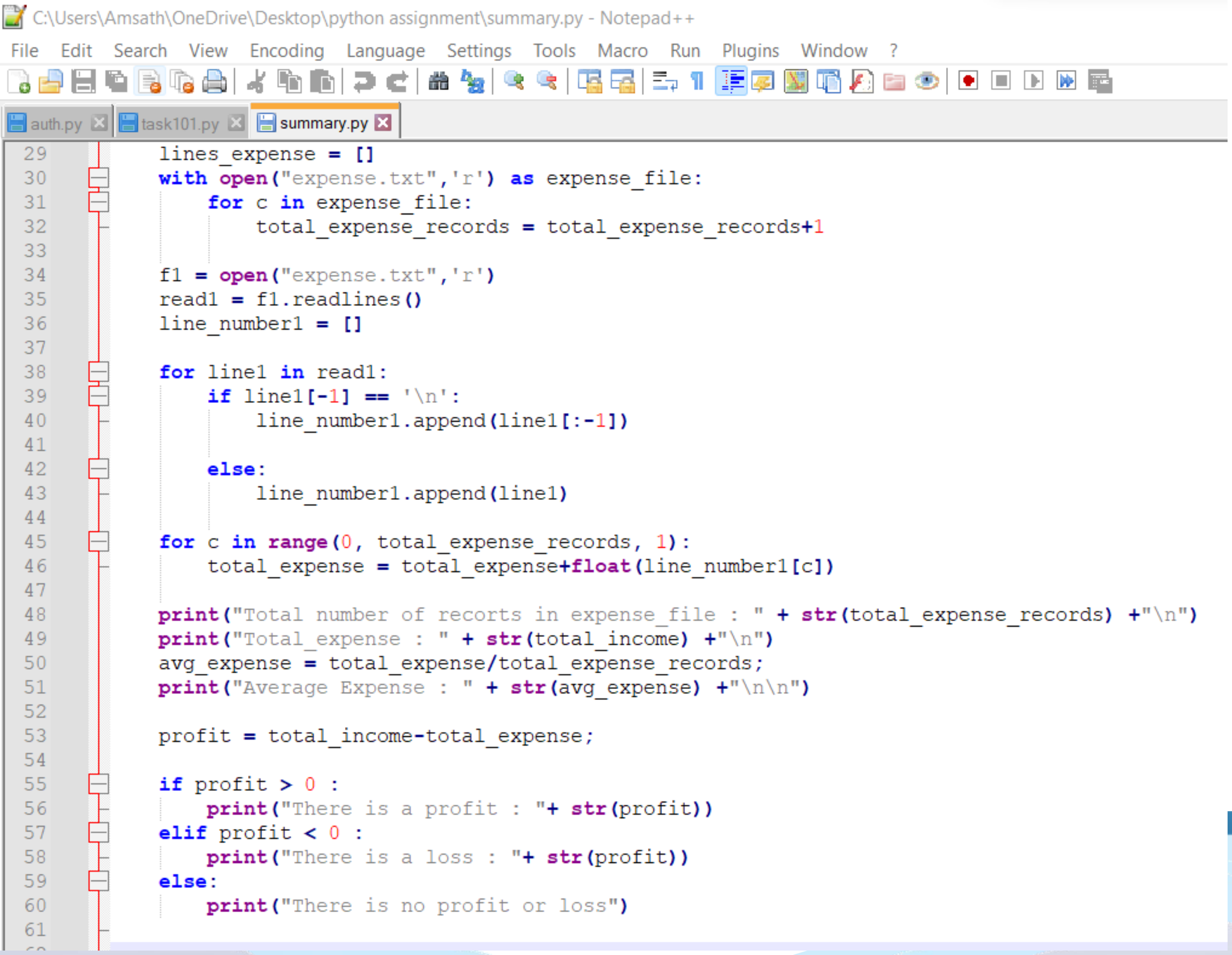
C:\Users\Amsath\OneDrive\Desktop\python assignment\task101.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
auth.py
task101.py
summary.py

```python
import datetime as dt

def daily_details():

    current = dt.datetime.now()
    s=current.strftime("%A %b %d %Y")
    print(s)

    income = float(input("Income : "))
    expense = float(input("Expense : "))

    fincome = open("Income.txt", 'a')
    fincome.write(str(income)+ "\n")
    fincome.close()

    fexpense = open("Expense.txt", 'a')
    fexpense.write(str(expense)+ "\n")
    fexpense.close()


```

Summary.py

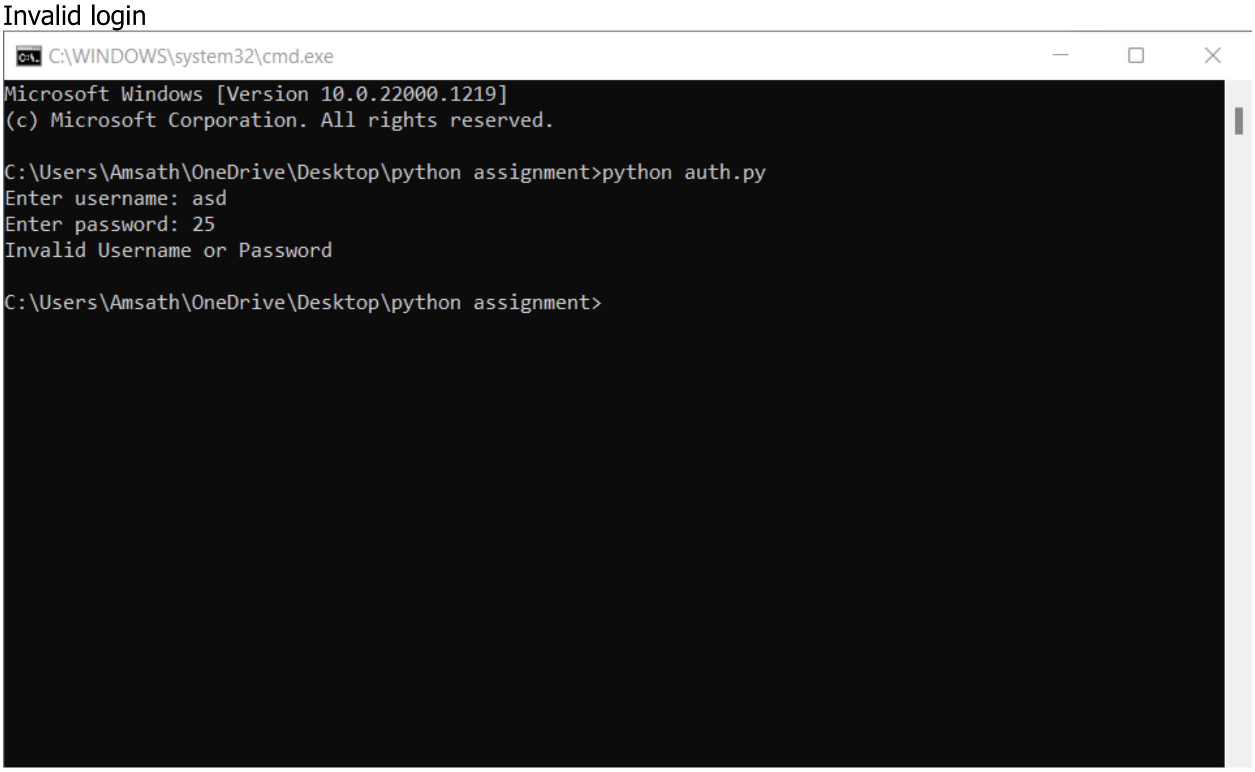

C:\Users\Amsath\OneDrive\Desktop\python assignment\summary.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
auth.py
task101.py
summary.py

```
def summary():
    total_income_records = 0
    total_income = 0
    with open("income.txt",'r') as income_file:
        for a in income_file:
            total_income_records = total_income_records+1

    f = open("income.txt",'r')
    read = f.readlines()
    line_number =[]

    for line in read:
        if line[-1] == '\n':
            line_number.append(line[:-1])

        else:
            line_number.append(line)

    for a in range(0, total_income_records,1):
        total_income = total_income+float(line_number[a])

    print("Total number of records in income_file : "+ str(total_income_records) +"\n")
    print("Total income : " + str(total_income) +"\n")
    avg = total_income/total_income_records;
    print("Average income : " + str(avg) +"\n\n")

    total_expense_records = 0
    total_expense = 0
    lines_expense = []
    with open("expense.txt",'r') as expense_file:
        for c in expense_file:
            total_expense_records = total_expense_records+1

    f1 = open("expense.txt",'r')
    read1 = f1.readlines()
    line_number1 = []

    for line1 in read1:
        if line1[-1] == '\n':
            line_number1.append(line1[:-1])

        else:
            line_number1.append(line1)
```


C:\Users\Amsath\OneDrive\Desktop\python assignment\summary.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
auth.py
task101.py
summary.py

```python
lines_expense = []
with open("expense.txt",'r') as expense_file:
    for c in expense_file:
        total_expense_records = total_expense_records+1

f1 = open("expense.txt",'r')
read1 = f1.readlines()
line_number1 = []

for line1 in read1:
    if line1[-1] == '\n':
        line_number1.append(line1[:-1])

    else:
        line_number1.append(line1)

for c in range(0, total_expense_records, 1):
    total_expense = total_expense+float(line_number1[c])

print("Total number of recorts in expense_file : " + str(total_expense_records) +"\n")
print("Total_expense : " + str(total_income) +"\n")
avg_expense = total_expense/total_expense_records;
print("Average Expense : " + str(avg_expense) +"\n\n")

profit = total_income-total_expense;

if profit > 0 :
    print("There is a profit : "+ str(profit))
elif profit < 0 :
    print("There is a loss : "+ str(profit))
else:
    print("There is no profit or loss")
```

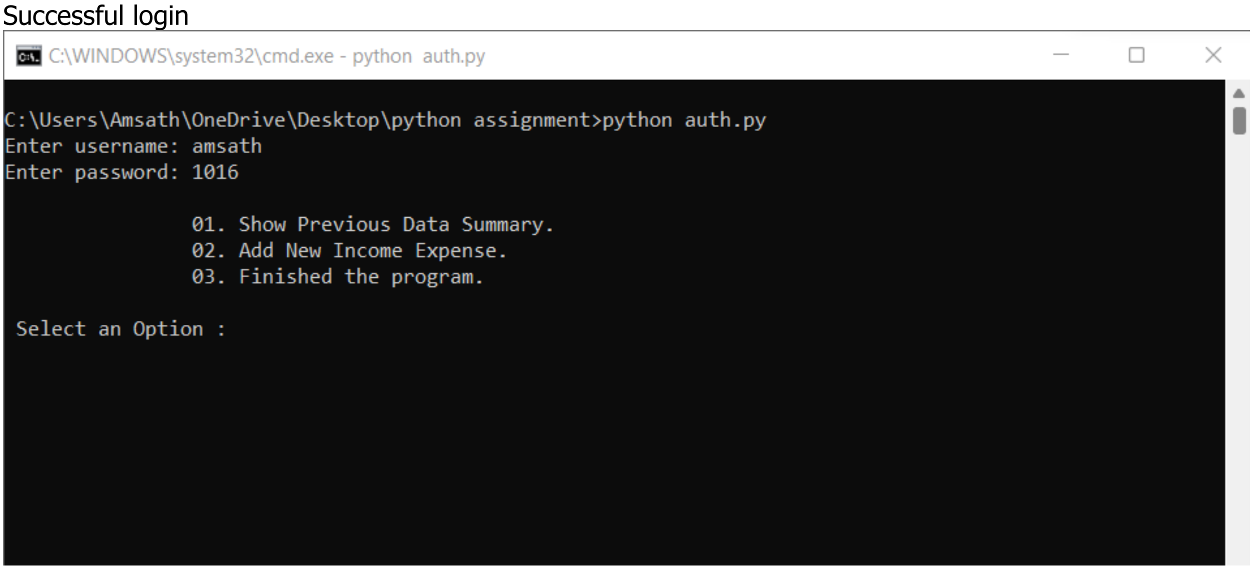
Invalid login



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amsath\OneDrive\Desktop\python assignment>python auth.py
Enter username: asd
Enter password: 25
Invalid Username or Password

C:\Users\Amsath\OneDrive\Desktop\python assignment>
```

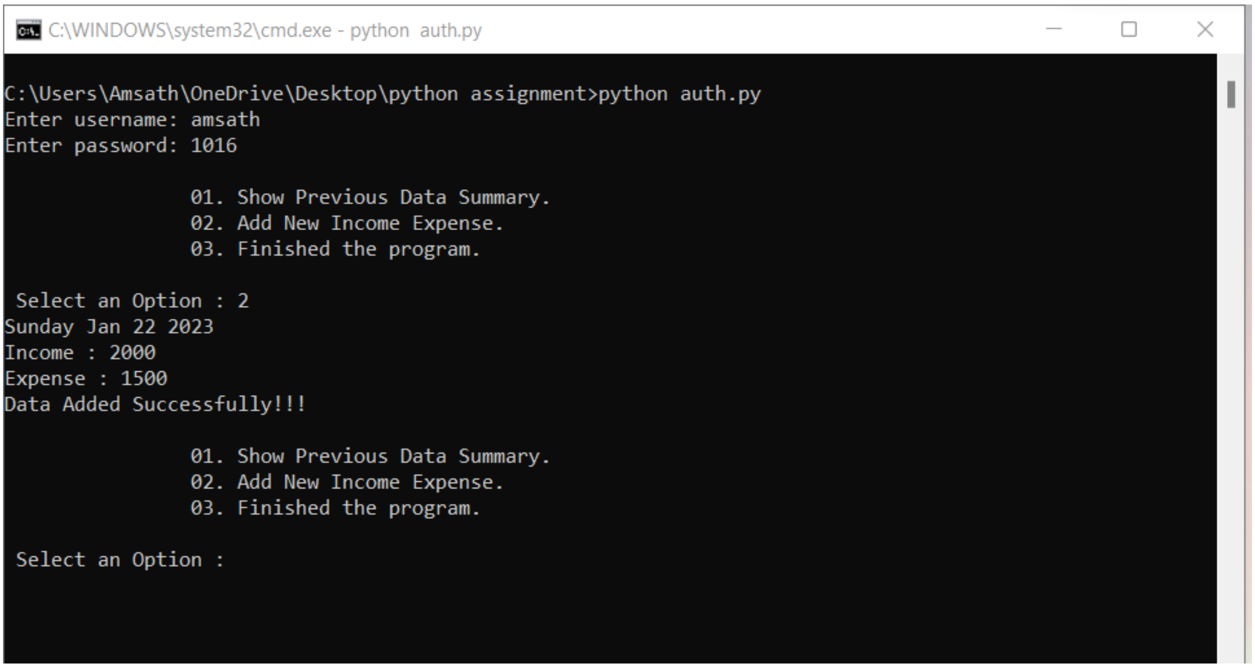
Successful login



C:\WINDOWS\system32\cmd.exe - python auth.py

```
C:\Users\Amsath\OneDrive\Desktop\python assignment>python auth.py
Enter username: amsath
Enter password: 1016

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option :
```

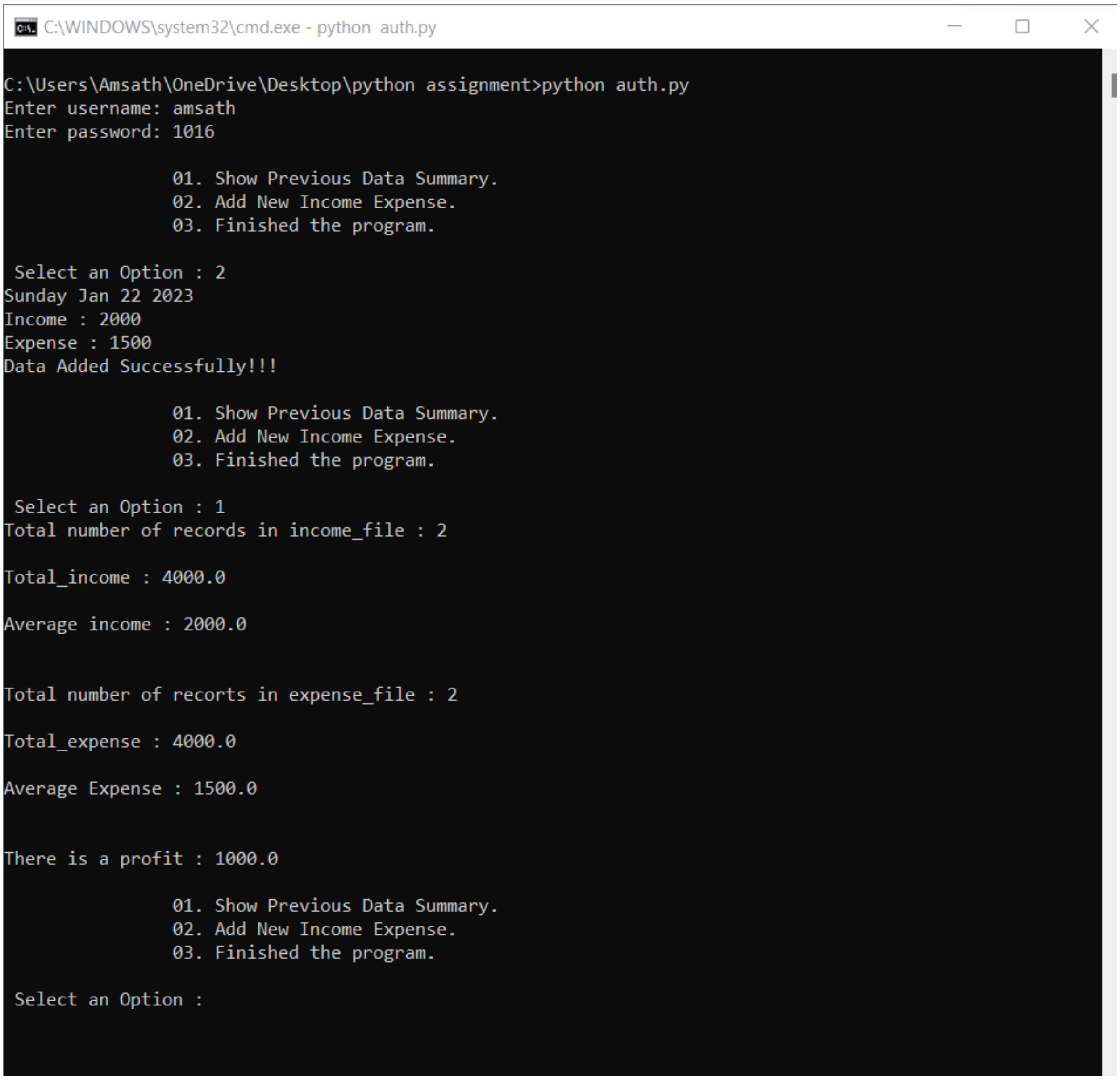
Add new data



C:\WINDOWS\system32\cmd.exe - python auth.py

```
C:\Users\Amsath\OneDrive\Desktop\python assignment>python auth.py
Enter username: amsath
Enter password: 1016

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 2
Sunday Jan 22 2023
Income : 2000
Expense : 1500
Data Added Successfully!!!

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option :
```

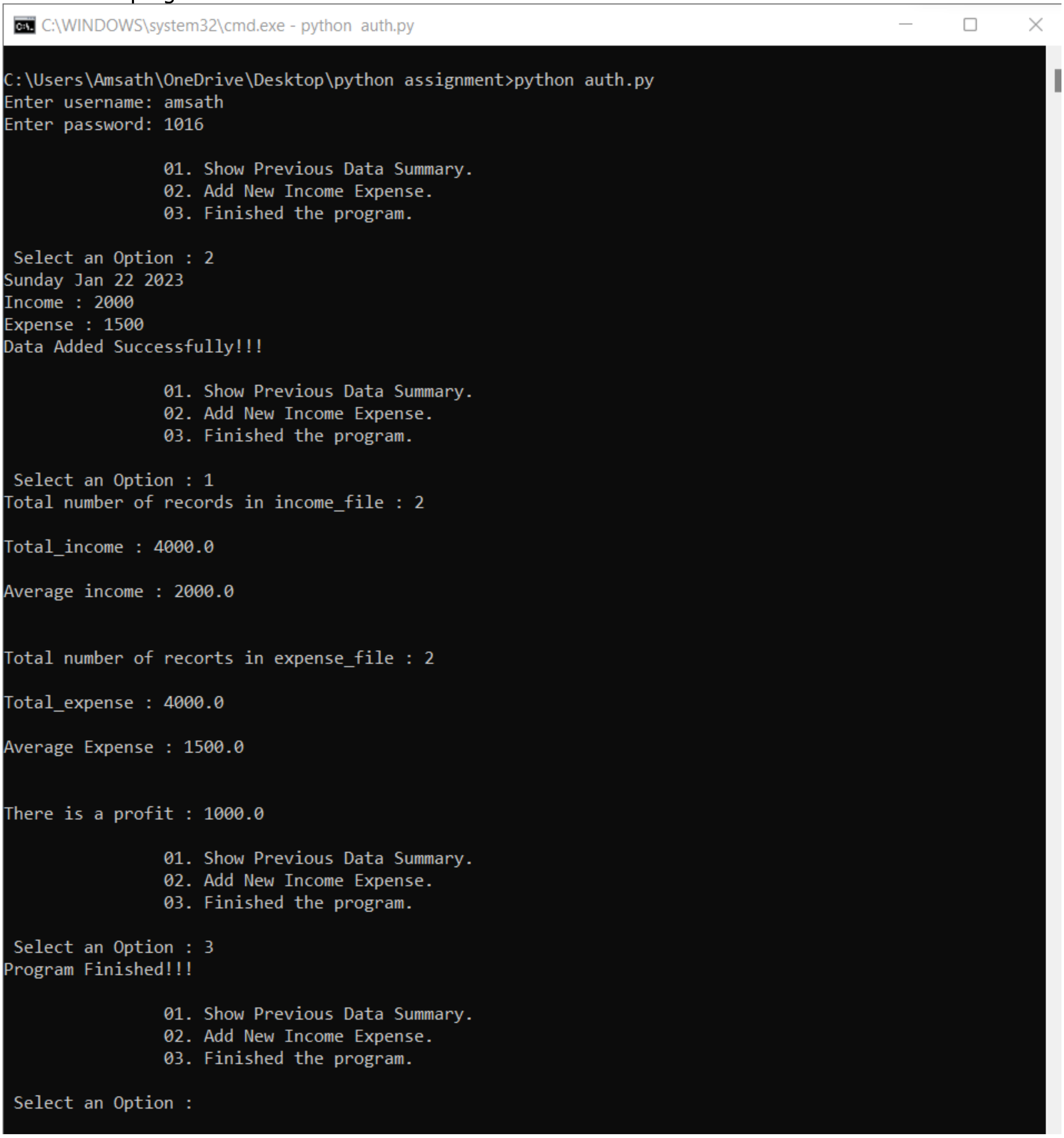
Show all records



C:\WINDOWS\system32\cmd.exe - python auth.py

```
C:\Users\Amsath\OneDrive\Desktop\python assignment>python auth.py
Enter username: amsath
Enter password: 1016

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 2
Sunday Jan 22 2023
Income : 2000
Expense : 1500
Data Added Successfully!!!

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 1
Total number of records in income_file : 2

Total_income : 4000.0

Average income : 2000.0


Total number of recorts in expense_file : 2

Total_expense : 4000.0

Average Expense : 1500.0


There is a profit : 1000.0

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option :
```

Finished the program



```
C:\Users\Amsath\OneDrive\Desktop\python assignment>python auth.py
Enter username: amsath
Enter password: 1016

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 2
Sunday Jan 22 2023
Income : 2000
Expense : 1500
Data Added Successfully!!!

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 1
Total number of records in income_file : 2

Total_income : 4000.0

Average income : 2000.0


Total number of recorts in expense_file : 2

Total_expense : 4000.0

Average Expense : 1500.0


There is a profit : 1000.0

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option : 3
Program Finished!!!

                01. Show Previous Data Summary.
                02. Add New Income Expense.
                03. Finished the program.

 Select an Option :
```

Task 4

4.1 Explain the debugging procedure and the debugging tools provided by the IDE you used to construct the method in Task 3.

Starting a debugger session is essentially the same as running the application normally. We don't have to do anything specific to begin a debugger session because the debugger is already attached in the background. If PyCharm allows us to run our program, we can use the same parameters to debug it.

- Visual Debugging: Because pdb is intimidating and the concept is hard to grasp, several programmers still debug via print statements. The Python debugging GUI in PyCharm makes using a debugger straightforward by giving the process a visual face. Starting out is easy, and moving on to the essential debugging features is simple.

- Debugging TDD, or test-driven development, involves experimentation as the tests are being created. To help you with your exploration, set breakpoints in the context you're examining.
- Breakpoints: All debuggers have breakpoints, but only a few of them have breakpoints that are particularly adaptive. Have you ever kept pressing the "continue" button until you got to the loop iteration when your issue appeared? There's no need for it with PyCharm's conditional breakpoints. Sometimes all you need to do while code is running is check a variable's value. Breakpoints in PyCharm can be set up such that they simply report a message rather than pausing your program. Because exceptions can ruin your day, PyCharm's debugger can trip on them even if you are unsure of their origin.
- Variable Values at a Glance: PyCharm will display all of your variable values inline within your code when it encounters a breakpoint. To make it straightforward to notice which values have changed since you last reached the breakpoint, modified values are underlined.

- Watches: To customize your variable view, add watches to it. Whether they are simple or complex, you will be able to view exactly what you want to see.

4.2 You should include in your report

I. Describe the coding convention you followed in Task 3's code.

- Function length is short
- No global variables are utilized
- Indentation is used
- Variable names are clear and intelligible
- The GOTO statement is never used.

II. Consider how the debugging process can be utilized to create applications that are more robust, secure, and reliable. It simply cannot. System defects are recreated, found, and described throughout the debugging process. It serves as a fix for problems that have already existed. Users in the field or testing are the two methods used to find these issues. This is what you do when a problem is acknowledged. Because of this, debugging is essential to improve application robustness even while it cannot help design more secure programs.
Since debugging is a reactive process, it is not a good way to boost security or robustness. Although testing is preferred, an application cannot be completely tested.
3. Analyze the reasons a coding standard is essential for both the team and the individual.

A coding standard makes sure that all developers involved in the project follow certain guidelines. The code is easy to read and maintains a sufficient level of consistency.
Consistency should be maintained when coding as it has a positive impact on program quality. Additionally, it should be made sure that all system layers adhere to the same coding standards and that they do not conflict with one another. The finished software code should be visible.
In terms of programming, think of the you that is right now and the you that is coming. Really. One factor is that you will always grow and learn, therefore for the most part, old code will smell terrible.
On the other side, as you investigate a problem, you'll learn a lot of recent, precise facts. A piece of that knowledge will be essentially lost as soon as the issue is resolved. In the future, you won't remember how you did it or what your strategy was. You'll also forget the rationale behind your choice of particular names and signatures.