



 slington college
(इस्लिङ्टन कलेज)

CS4051N Fundamental Of Computing

60% Individual Coursework - 2

2023-24 Spring

Student Name: Sikum Hangma Madi

London Met ID: 22085627

College ID: NP01CP4S230077

Group: C20

Assignment Due Date: Friday, August 25, 2023

Assignment Submission Date: Friday, August 4, 2023

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

1. Introduction	1
1.1 Objective and Goals.....	1
1.2 Tools used:.....	2
1) Python.....	2
2) Thonny	2
3) Ms Word.....	3
4) Draw.io.....	5
2. Discussion and Analysis.....	6
2.1 Algorithm	6
2.2 Flowchart	8
2.3 Pseudocode.....	11
Main:.....	11
Read:	12
Operation:.....	15
Write:	20
1. Data Structure	23
3.1 Implementation of Data Structure in my course work	23
3.2 Primitive Data type.....	25
3.3 Collective Data type.....	25
4. Testing	26
4.1 Test-1	26
4.2 Test-2.1	27
4.2 Test-2.2.....	29
4.3 Test-3.....	31
4.4 Test-4.....	33
4.5 Test-5.....	35
5. Conclusion	36
5.1 Summary.....	36
5.2 limitation of the project implementation.....	36
5.2 Research and findings	37
6. Reference	37
Bibliography	37

Appendix	38
Main file	38
read file	39
Operation file	41
Write file.....	49

Figure 1 Python	2
Figure 2 Thonny	3
Figure 3 Ms Word.....	4
Figure 4 Draw.io	5
Figure 5 flowchart(a)	10
Figure 6 flowchart(b)	10
Figure 7 Implementation of Dictionary data structure	24
Figure 8 Test-1.1, implementation of try and except block	27
Figure 9 Test 1.2, implementation of Try except block	27
Figure 10 Test-2.1 selecting rent and providing negative value	28
Figure 11 Test-2.1 selecting return and providing negative value	29
Figure 12 Test-2.2 selecting rent and providing non-existence value	30
Figure 13 Test-2.2 selecting return and providing non-existing value	30
Figure 14 Test-3 generating rent invoice with deduction of quantity of equipment	32
Figure 15 Test-4 generating return invoice with increase of quantity of equipment	34
Figure 16 Test-5 Update in inventory	35
Figure 17 Test-5 Update of inventory in .txt file	36

Table 1 Test-1 implementation of try and except	26
Table 2 Test-2.1 showing selection of renting returning providing negative number	28
Table 3 Test-2.2 selecting renting and returning providing non-existing number	29
Table 4 Test-3 generating file for renting equipment	31
Table 5 Test-4 generating file for returning equipment	33
Table 6 Test-5 updating inventory of equipment after renting and returning	35

1. Introduction

The coursework required to develop a system with detailed information guidelines to the to the customers by using the python programming language. The assigned coursework contains 60% of the total weightage in the gradings of the final result of our second semester. Here, we have used the assistance of Python and Microsoft Word for this coursework for detailed coding and documentation.

1.1 Objective and Goals

The main objective of the given coursework was to develop Python bases Rental Management System for Equipment Rental shop which has the function to rent, return and generate a bill while also maintaining the information of Equipment in txt file. The project consists of four files, including main.py for core functionality and read_write_operations.py for file management. This system will make it easier for users to sign up, keep track of the inventory of products, and rent and return items. Based on the products and length of the rental, it will provide detailed invoices. The project seeks to speed rental activities, assure data integrity through file operations, and deliver an effortless, error-handled experience with an emphasis on security, user-friendliness, and scalability. The system's functioning will be validated, and its growth potential will be increased, through thorough testing and documentation.

1.2 Tools used:

1) Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy prioritizes code readability and heavily employs indentation.

Python has dynamic typing and garbage collection. In addition to structured programming (particularly this), it supports a number of other programming paradigms, such as procedural, object-oriented, and functional programming. It is frequently referred to as a "batteries included" language because of its large standard library. (Gosling) It is used in this coursework for developing the application. All the coding part is done here.

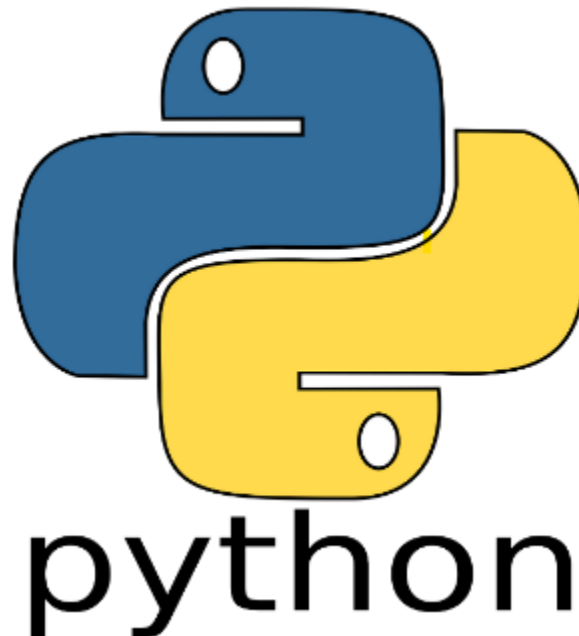


Figure 1 Python

2) Thonny

Thonny is free and open-source integrated development integrated area(IDE) that is designed for beginners. It was created by Aivar Annamaa, an Estonian

programmer. It supports different ways of stepping through code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap. (Tartu, 2014 to 2018) Thonny comes with Python 3.10 built in, so just one simple installer is needed and you're ready to learn programming. (You can also use a separate Python installation, if necessary.) The initial user interface is stripped of all features that may distract beginners. While I was suffering with lots of small errors, Thonny helped me with debugging process with its simple features it was really easy to use. (educative.io, 2019)



Figure 2 Thonny

3) Ms Word

Microsoft created the word processing program known as Microsoft Word. It was initially made available on October 25, 1983 for Xenix systems under the name Multi-Tool Word. Later versions were created for a variety of other operating systems, including SCO Unix (1990), Microsoft Windows (1989), Atari ST (1988), OS/2 (1989), IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS (1985), AT&T UNIX PC (1985), and macOS (1990). (2001). Microsoft Word versions prior to 2013 can be used on Linux by using Wine. (techopedia, Ms word, 2022) Word is available commercially as a stand-alone product or as a part of

the Microsoft Office software suite, which can be purchased as a permanent license or as a monthly subscription to Microsoft 365. Word can also be purchased along with Windows. Ms word has helped me throughout the documentation process.



Figure 3 Ms Word

4) Draw.io



Figure 4 Draw.io

Its interface can be used to construct diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams. Diagrams.net (formerly draw.io is a free and open source cross-platform graph sketching program written in HTML5 and JavaScript.

Diagrams.net is available for Linux, macOS, and Windows as an offline desktop application and as a cross-browser web app online. Its offline application was created with the Electron technology. The web program can be opened from and saved on the local hard drive without requiring an online login or registration. PNG, JPEG, SVG, and PDF are supported export and storage formats that can be downloaded. Additionally, it integrates with cloud storage services like GitHub, GitLab.com, OneDrive, Dropbox, and Google Drive. Additionally, a plugin for web embedding is available. (sin)

2. Discussion and Analysis

2.1 Algorithm

An algorithm is a set of guidelines for resolving a dilemma or carrying out a task. A recipe, which consists of detailed directions for creating a dish or meal, is a typical illustration of an algorithm. Algorithms are used by every computerized equipment to carry out its operations in the form of hardware- or software-based routines.

STEP 1: Start

STEP 2: Display welcome message

STEP 3: Display the selection option to select the operation for renting, returning or exit.

STEP 4: Get user's input

STEP 5: If user enters 1 go to Step 6 , if user enters 2 go to Step 14 , if user enters 3 to exit go to Step 23

STEP 6: Display the available item details for renting.

STEP 7: Ask the user to enter the SN of equipment for renting.

STEP 8: Ask the user to enter the quantity of the equipment

STEP 9: Ask the user if he wants to rent any more itm: Y/N

STEP 10: If the users enters Y go to step 6

STEP 11: If the users enters N go to step 11

STEP 12: Generate bill

STEP 13: Go to step 3.

STEP 14: Display the available item details for returning.

STEP 15: : Ask the user to enter the SN of equipment for returning.

STEP 16: Ask the user to enter the quantity of the equipment

STEP 17: Ask the user to how many days you were late returning.

STEP 18: Ask the user if they want to continue returning: Y/Enter

STEP 19: If the user enters Y go to step 15.

STEP 20: If the user enters Enter go to step 21

STEP 21: Generate Bill for returning

STEP 22: Go to Step 3

STEP 23: Display Thankyou Message

STEP 24: End program

2.2 Flowchart

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. It's a common process analysis tool and one of the seven basic tools.

Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements. (asp.org)

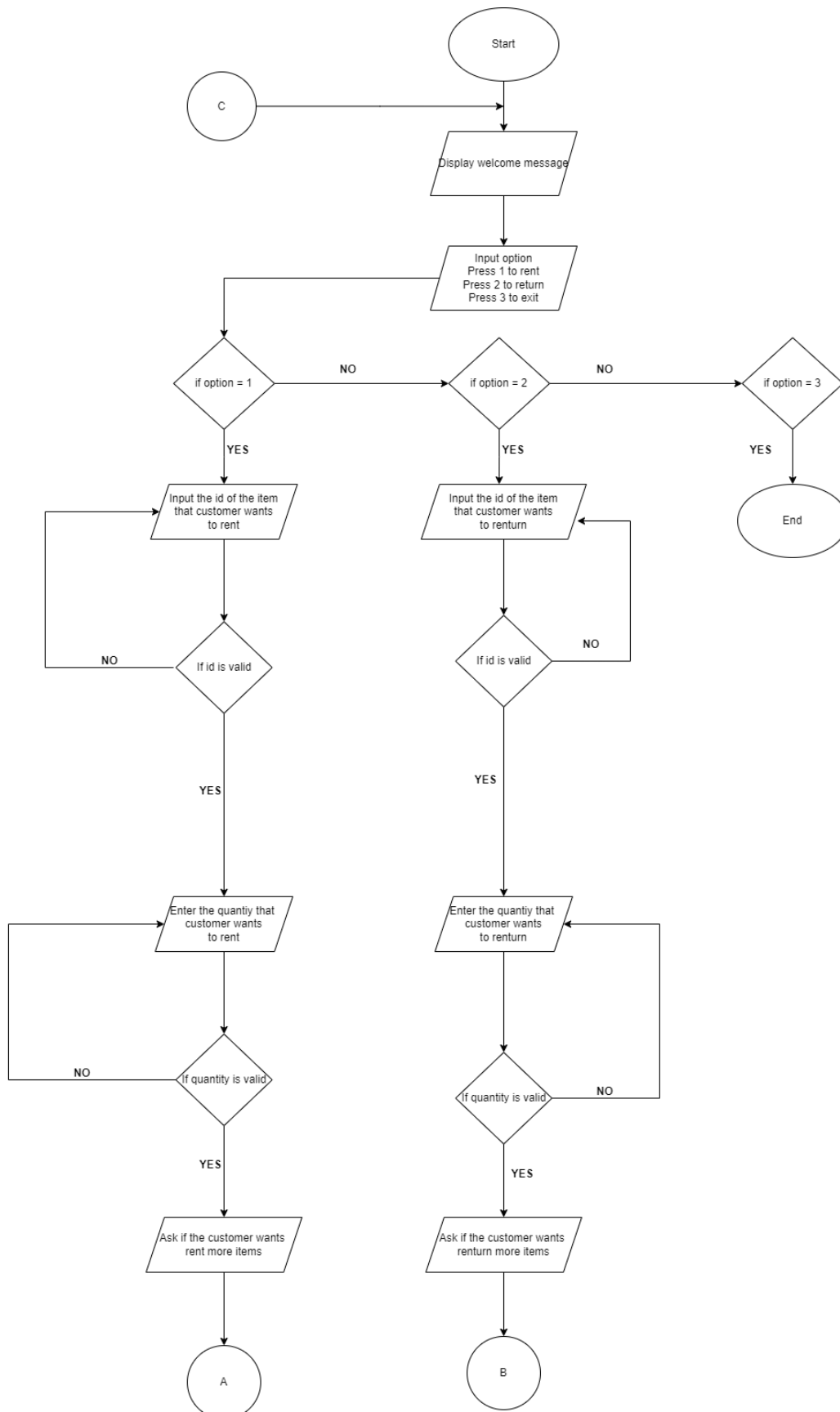


Figure 5 flowchart(a)

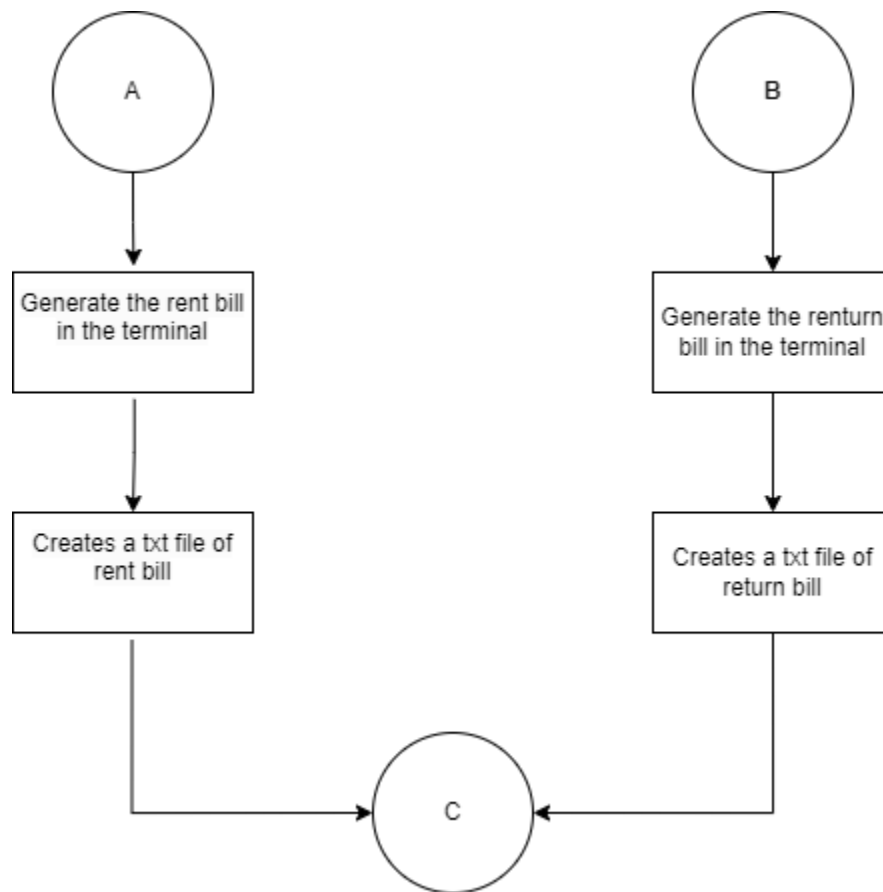


Figure 6 flowchart(b)

2.3 Pseudocode

Rather than using a programming language, pseudocode (pronounced SOO-doh-kohd) is a precise yet accessible description of what an algorithm or computer program must do. Sometimes a stage in the process of creating a program is described in depth using pseudocode. It enables lead programmers or designers to define the concept in great detail and gives programmers a thorough template for the subsequent step of writing code in a particular programming language. (techopedia, Pseudocode, 2019)

The team of designers and programmers can review pseudocode to make sure that actual programming is likely to adhere to design standards because it is thorough yet comprehensible. Finding faults early on in the development process is less expensive than doing it later. The acceptable pseudocode is then recast in a programming language's vocabulary and grammar. Sometimes approaches based on computer-aided software engineering are combined with pseudocode. It is feasible to create programs that translate from one pseudocode language to another into another. Here, are the pseudocode of the codes used in this project:

Main:

IMPORT datetime

IMPORT read

IMPORT operation

CALL welcome_message() function

WHILE loop is True

TRY

 userinput=display_opt()

IF userinput == 1

 Checking rent_fuction()

```
        PRINT "Your item has been rented"
    ELSEIF userInput == 2
        Checking return_fuction()
        PRINT "Your item has been returned"
    ELSEIF userInput == 3
        PRINT "Thankyou for visiting us"
        Loop is False
    ELIF userInput GREATER than 3 OR userInput LESS than 0
        PRINT "Enter the correct option"
EXCEPT ValueError
    PRINT "Invalid Entry. Please Enter a valid numerical option."
```

Read:

```
IMPORT datetime
```

```
FUNCTION welcome_message()
```

```
    PRINT .....-
    PRINT "Welcome to our System. I hope you have a good time here"
    PRINT .....-
```

```
FUNCTION read_txt()
```

```
    OPEN file "equipment.txt" for reading as file
    INITIALIZE an empty dictionary myDictionary
    INITIALIZE a variable itemSN with value 1
```

```
    FOR each line in file
```

```
        REMOVE newline character from the line
        SPLIT the line using comma and store it as a list
```


ADD the list as a value to myDictionary with itemSN as key

INCREMENT itemSN by 1

CLOSE the file

RETURN myDictionary

END Function

FUNCTION bill_1()

PRINT [Download](#) [Print](#)

PRINT "You must enter your personal info for billing purpose"

PRINT

INPUT name from the user

INPUT phone number from the user

RETURN name and number

END Function

FUCNTION display_equipment()

```
PRINT "~~~~~Equipment DETAILS~~~~~"
```

PRINT

```
PRINT "S.N. \t Equipments Name \t \t \t Brand \t \t \t Price \t \t Quantity"
```

SET s as 1

OPEN file "Equipment.txt" for reading as file

FOR each line in file

PRINT "-----"

PRINT s, "\t" + Replace commas in line with tabs

INCREMENT s by 1

CLOSE the file

FUCNTION valid_sn1()

PRINT "- "

INPUT SN of the item from the user for renting

PRINT "- "

RETURN valid_sn1

END Function

FUCNTION valid_sn2()

PRINT "- "

INPUT SN of the item from the user for returning

PRINT "- "

RETURN valid_sn2

END Function

FUNCTION display_opt():

PRINT "Here are some available options: "

PRINT "- "

PRINT "\t Press 1 to rent an equipment"

PRINT "\t Press 2 to return"

PRINT "\t Press 3 to exit"

PRINT "- "

INPUT selecting option from the user

RETURN userinput

END Function

Operation:**IMPORT** datetime**IMPORT** read**IMPORT** write**FUCNTION** rent_fucntion()

myDictionary equals read_txt()

SET user_purchase equipments to emprty list**WHILE** True **CALL** display equipments() function **SET** valid_sn to valid_sn1() function **WHILE** valid_sn **LESSER THAN OR EQUAL TO** 0 **OR** valid_sn
 GREATER than length of myDictionary **PRINT** "Ivalid Entry. Please provide a valid equipment sn" **INPUT** integer SN for return **INPUT** integer quantity fir return **WHILE** user_quantity is **GREATER** than or **EQUAL TO** 0 **OR**
 user_quantity is **GREATER** than integer
 get_quality_of_selected_equipment **PRINT** "Dear Admin, the equipment you are looking for is not
 available at the moment. Please check the table and insert the
 equipment" **TRY** **INPUT** valid positive quantity **EXCEPT** ValueError **PRINT** "Please enter a valid integer value" **CONTINUE** **PRINT**-

myDictionary[valid_snn][3] equals converting string into int where
(myDictionary[valid_snn][3])+ (user_quantity)

OPEN the file "equipment.txt" for writing as file

FOR each values in myDictionary.values()

WRITE newline characters to file

CLOSE the file

Name_of_product equals to 0 index of myDictionary

Quantity_selected_by_user equals to user_quantity

Unit_price equals to 2nd index of myDictionary

Price_of selected_equipment equals remove "\$" from 2nd index of myDictionary

Total_price equals to convert price_of_selected_equipment to integer * convert quantity_selected_by_user to integer

APPEND Name_of_product, Quantity_selected_by_user, Unit_price, Total_price to user_purchased equipments

INPUT user_req if they want to rent again

IF user_req id 'n'

BREAK

ELSEIF user_req is not 'y'

CONTINUE

DECLARE total as 0

FOR each l in user_purchased equipments

Grand_total equals to total

Today_date_and_time equals to get current date and time

PRINT Sikum's Equipment shop

PRINT "Kamalpokhari, Kathmandu | Contact No: 9855854555"

PRINT-

PRINT "Equipments details are"

PRINT Name of the Costumer:"

PRINT "contact number :"

PRINT "equipment name"

FOR each l in user_purchased_equipment

PRINT "grand total"

PRINT "Note. Fine cost will be added to grand total"

RETURN name, phone, today_date_and_time,
user_purchased_equipmemts, grand_total from write_rent

END Function

FUCNTION return_fuction()

myDictionary equals read_txt()

SET user_returned equipments to emprty list

Fine equals to 0

Fineday equals to 0

DECLARE more as True

WHILE more equals to True

CALL display equipments() function

SET valid_snn to valid_sn2() function

WHILE valid_snn **LESSER THAN OR EQUAL TO** 0 **OR** valid_snn
GREATER than length of myDictionary

PRINT "Invalid Entry. Please provide a valid equipment sn"

INPUT integer SN for return

INPUT integer quantity for return

WHILE user_quantity is **GREATER** than or **EQUAL TO** 0 **OR**
user_quantity is **GREATER** than integer
get_quality_of_selected_equipment

PRINT "Dear Admin, the equipment you are looking for is not available at the moment. Please check the table and insert the equipment"

TRY

INPUT valid positive quantity

EXCEPT ValueError

PRINT "Please enter a valid integer value"

CONTINUE

PRINT-

myDictionary[valid_snn][3] equals converting string into int where
(myDictionary[valid_snn][3]) + (user_quantity)

OPEN the file "equipment.txt" for writing as file

FOR each values in myDictionary.values()

WRITE newline characters to file

CLOSE the file

Name_of_product equals to 0 index of myDictionary

Quantity_selected_by_user equals to user_quantity

Unit_price equals to 2nd index of myDictionary

Price_of_selected_equipment equals remove "\$" from 2nd index of myDictionary

Total_price equals to convert price_of_selected_equipment to integer * convert quantity_selected_by_user to integer

TRY

INPUT number of days you were late in rentDays

IF rentDays is **LESS** than 1

PRINT "Invalid rented Days. Please Enter again"

ELSE

IF rentDays is **LESS** than or **EQUAL** to 5

```
Fine equals to 0
ELIF rentDays % 5 != 0
    Fineday = (int(rentDays//5)+1)
    Fine = int(fineday/5)*int
ELSE
    Fine equals to rentDays – 5 *
    int(price_of_selected_equipment)
INPUT 'Y' for returning more equipment or enter
    'Enter'
IF input equals to 'Y'
    more equals to True
ELSE
    Total equals to 0
    FOR I in user_returned equipments
        Total = total _ int(i[3])
        Grand_total equals to sum of total and
        fine
EXCEPT ValueError
    PRINT "Invalid days. Please try again"

APPEND Name_of_product, Quantity_selected_by_user,
Unit_price, Total_price, rentDays,fine to user_returned equipments

DECLARE total as 0
FOR each I in user_purchased equipments
    Grand_total equals to total
    Today_date_and_time equals to get current date and time

PRINT Sikum's Equipment shop
```

PRINT "Kamalpokhari, Kathmandu | Contact No:
9855854555"

PRINT -----

PRINT "Equipments details are"

PRINT Name of the Costumer:"

PRINT "contact number :"

PRINT "equipment name"

FOR each I in user_purchased_equipment

PRINT "grand total"

PRINT "Note. Fine cost will be added to grand total"

RETURN name, phone, today_date_and_time, user_returned equipments,
grand_total from write_rent

END Function

Write:

IMPORT operation

FUNCTION write_rent(name,phone,today_date_and_time,
user_purchased equipments,grand_total)

OPEN a file named name + phone + ".txt" in write mode as file

WRITE "-----"

WRITE ""Sikum's Equipment Rental Shop" to file

WRITE "\\t\\t\\t Kamalpokhari, Kathmandu | contact no. 9855854555" to file

WRITE "-----"

WRITE "Equipment Details are :" to file

WRITE "-----"

WRITE "Name of the customer :" + name to file

WRITE "Contact Number :" + phone to file

WRITE "Date and time of Purchase :" + today_date_and_time to file


```

WRITE "-----"
WRITE "Purchase Details are :" to file
WRITE "Equipment Name \t\t Total Quantity \t\t Unit Price \t\t Total" to file
WRITE "-----"
FOR each equipment in user_purchased equipments
    WRITE equipment of index 0 + "\t\t + equipment of index 1" + "\t\t\
    +equipment of index 2 + "\t\t\t" + "$" + equipment of index 3 to file
    WRITE "-----"
WRITE "Grand Total : $" + grand_total to file
WRITE "Note: fine cost will be added to grand total if there is delay in return"

FUNCTION write_return(name,phone,today_date_and_time,
user_returned equipments,grand_total)
    OPEN a file named name + phone + ".txt" in write mode as file
    WRITE "-----"
    WRITE ""Sikum's Equipment Rental Shop" to file
    WRITE "\t\t\t\t\t Kamalpokhari, Kathmandu | contact no. 9855854555" to file
    WRITE "-----"
    WRITE "Equipment Details are :" to file
    WRITE "-----"
    WRITE "Name of the customer :" + name to file
    WRITE "Contact Number :" + phone to file
    WRITE "Date and time of Purchase :" + today_date_and_time to file
    WRITE "-----"
    WRITE "Purchase Details are :" to file
    WRITE "Equipment Name \t\t Total Quantity \t\t Unit Price \t\t Total" to file
    WRITE "-----"
    FOR each equipment in user_returned equipments

```

WRITE equipment of index 0 + "\t\t + equipment of index 1" + \t\t\
+equipment of index 2 + "\t\t\t" + "\$" + equipment of index 3 to file

WRITE "-----"

WRITE "Grand Total : \$" + grand_total to file

WRITE "Note: fine cost will be added to grand total if there is delay in return"

1. Data Structure

Python has a wide range of data types, which makes working with groupings of data much easier. Simpler. Dictionaries, lists, tuples, and strings are the different forms of data in Python. Multiple data types make up these data structures. In a nutshell, strings are text fragments while tuples and lists are ordered collections of discrete data objects. Dictionaries are collections of key-value pairs. For lists, tuples, and strings, the method for acquiring access to data in order is the same.

3.1 Implementation of Data Structure in my course work

I've included a dictionary as my main data structure in course work. I've developed a flexible and organised method of information storage by using dictionaries. Each dictionary represents a single item, with the key being its own item identifier. This enables quick and easy access to each item's dictionary, which contains key-value pairs that represent the item's name, description, and rental rate. My rental management system's operating efficiency is significantly improved by this implementation option, which streamlines data retrieval, changes, and additions. The usage of dictionaries makes it easier to read the code, ensures that the information is organised and simple to access, and helps the project to succeed overall.

```

def rent_function():
    myDictionary = read_txt()
    user_purchased equipments = []

    while True:
        display equipments()
        valid_sn = valid_sn1()
        # Valid ID
        while valid_sn <=0 or valid_sn > len(myDictionary):
            print("Invalid Entry.Please provide a valid equipment ID !!!")
            print("\n")
            valid_sn = int(input("Please enter S.N. of the Equipment you want to rent: "))

        print("-----")
        user_quantity = int(input("Please provide the number of quantity of the equipment you want to rent: "))
        print("-----")
        print("-----")

        #Valid quantity
        get_quantity_of_selected_equipment = myDictionary[valid_sn][3]
        while user_quantity <=0 or user_quantity > int(get_quantity_of_selected_equipment):
            print("Dear Admin, the Equipment you are looking for is not available at the moment. Please check the table .")
            print("\n")

            try:
                user_quantity = int(input("Please provide a valid positive quantity: "))
            except ValueError:
                print("Please enter a valid integer value.")
                continue
            print("-----")

        #Update the text file
        myDictionary[valid_sn][3] = int(myDictionary[valid_sn][3]) -int(user_quantity)
        file = open("equipment.txt","w")

```

Figure 7 Implementation of Dictionary data structure

For my coursework, I decided to take advantage of dictionaries' adaptability and organising capability. I'm able to handle data well by using dictionaries, especially item inventory details, which fit naturally with the key-value structure of dictionaries. This structure offers a simple method for storing and retrieving item information. Because dictionaries are naturally flexible, I can manage complex data associations with ease, resulting in quick and effective information retrieval and manipulation. My rental management system's efficiency and cohesion have substantially increased after I switched to dictionaries, improving both its utility and readability.

Uses of 2d list:-

- 2d list uses tables and grids where you have rows and columns to organize data.

- It is used to represent matrices in mathematics which allows for operation like addition, multiplications and transformation.

3.2 Primitive Data type

The essential building blocks of programming languages are primitive data types, commonly referred to as elementary or fundamental data types. They stand in for the most basic values that a computer language can directly operate with. These data types often have fixed memory sizes and are predefined by the language itself.

Some of the primitive data types are:-

- Integer – represents whole number both positive and negative
- Float – represents real numbers with decimal
- Character- represents single characters like letters, digits as a symbol.
- Boolean- represents true or false value which are used as logical expressions

3.3 Collective Data type

Data types that group numerous values together are referred to as collective data types, composite data types, or structured data types. They give you the ability to build more intricate structures to represent data that has numerous properties or constituents.

Some of the collective data types are:-

- Array – ordered collection of elements that are of same data type
- List – dynamic type of array where size can be customized
- Tuple – ordered collection of elements like array but can have different data type
- Dictionary/ map – stores key-value as in pairs allowing it to associate data with specific key for its retrieval
- Set – collection of distinct elements with no defined order

4. Testing

4.1 Test-1

Objective	To show the implementation of try and except block
Action	provide invalid input and show message
Expected Result	Message appeared should appear
Actual Result	message appeared
Conclusion	The test was successful.

Table 1 Test-1 implementation of try and except

```

.....
                Welcome to our system. I hope you have a good time here
.....

Here are some available options:
-----
    Press 1 to rent an equipment
    Press 2 to return
    Press 3 to exit
-----

Please select an option: a
Invalid Entry. Please Enter a valid numerical option
Here are some available options:
-----
    Press 1 to rent an equipment
    Press 2 to return
    Press 3 to exit
-----

Please select an option:

```

Figure 8 Test-1.1, implementation of try and except block

```

-----
Please select an option: 1
                ~~~~~Equipment DETAILS~~~~~
.....
S.N.   Equipments Name                Brand                Price                Quantity
-----
1      Velvet Table Cloth              Saathi                $8                   20
-----
2      Microphone Set                  Audio Technica         $189                 15
-----
3      Disco Light Set                 Sonoff                 $322                 24
-----
4      7.1 Surround Sound Speaker Set  Dolby                  $489                 4
-----
5      Dinner Table 8x5                 Panda Furnitures       $344                 8
-----

Please provide the SN of the item you want to rent: a
Invalid Entry. Please Enter a valid numerical option
Here are some available options:

```

Figure 9 Test 1.2, implementation of Try except block

4.2 Test-2.1

Objective	To show the selection of renting and returning of equipment.
Action	-Providing negative number
Expected Result	Loop should return to the selecting options showing error message
Actual Result	Loop returned to the selecting options showing error message
Conclusion	Test was successful

Table 2 Test-2.1 showing selection of renting returning providing negative number

```

-----
Please select an option: 1
          ~~~~~Equipment DETAILS~~~~~
-----
S.N.      Equipments Name      Brand      Price      Quantity
-----
1         Velvet Table Cloth    Saathi      $8         20
-----
2         Microphone Set        Audio Technica    $189      15
-----
3         Disco Light Set       Sonoff       $322      24
-----
4         7.1 Surround Sound Speaker Set    Dolby      $489      4
-----
5         Dinner Table 8x5      Panda Furnitures    $344      8
-----

Please provide the SN of the item you want to rent: 1
-----
Invalid Entry.Please provide a valid equipment ID !!!

```

Figure 10 Test-2.1 selecting rent and providing negative value


```

-----
Please select an option: 2
          ~~~~~Equipment DETAILS~~~~~
-----
S.N.      Equipments Name      Brand      Price      Quantity
-----
1         Velvet Table Cloth    Saathi     $8         20
-----
2         Microphone Set        Audio Technica $189       15
-----
3         Disco Light Set       Sonoff     $322       24
-----
4         7.1 Surround Sound Speaker Set Dolby     $489       4
-----
5         Dinner Table 8x5      Panda Furnitures $344       8
-----

Please provide the SN of the item you want to return: -6
-----
Invalid Entry.Please provide a valid equipment's S.N. !!!
-----

```

Figure 11 Test-2.1 selecting return and providing negative value

4.2 Test-2.2

Objective	To show the selection of renting and returning of equipment.
Action	-Providing non- existing number
Expected Result	Loop should return to the selecting options showing error message
Actual Result	Loop returned to the selecting options showing error message
Conclusion	Test was successful

Table 3 Test-2.2 selecting renting and returning providing non-existing number

```

-----
Please select an option: 1
          ~~~~~Equipment DETAILS~~~~~
-----
S.N.    Equipments Name                Brand                Price                Quantity
-----
1       Velvet Table Cloth              Saathi                $8                   20
-----
2       Microphone Set                  Audio Technica        $189                 15
-----
3       Disco Light Set                 Sonoff                $322                 24
-----
4       7.1 Surround Sound Speaker Set  Dolby                 $489                 4
-----
5       Dinner Table 8x5                Panda Furnitures      $344                 8
-----
Please provide the SN of the item you want to rent: 9
-----
Invalid Entry.Please provide a valid equipment ID !!!

```

Figure 12 Test-2.2 selecting rent and providing non-existence value

```

-----
Please select an option: 2
          ~~~~~Equipment DETAILS~~~~~
-----
S.N.    Equipments Name                Brand                Price                Quantity
-----
1       Velvet Table Cloth              Saathi                $8                   20
-----
2       Microphone Set                  Audio Technica        $189                 15
-----
3       Disco Light Set                 Sonoff                $322                 24
-----
4       7.1 Surround Sound Speaker Set  Dolby                 $489                 4
-----
5       Dinner Table 8x5                Panda Furnitures      $344                 8
-----
Please provide the SN of the item you want to return: 10
-----
Invalid Entry.Please provide a valid equipment's S.N. !!!

```

Figure 13 Test-2.2 selecting return and providing non-existing value

4.3 Test-3

Objective	To generate a file for renting equipment(multiple equipment).
Action	2 equipment were rented.
Expected Result	Equipment should be rented and bill should be displayed with deduction of quantity in the inventory.
Actual Result	Equipment was rented and bill was displayed with deduction of quantity in the inventory.
Conclusion	Test was successful

Table 4 Test-3 generating file for renting equipment

```

Sikum's Equipment shop

Kamalpokhari, Kathmandu | Contact No: 9855854555

-----
Equipment Details are:
-----
Name of the Costumer:sikum
Contact number: 98334455
Date and time of purchase: 2023-08-24 09:06:55.793905
-----

Purchase Detail are:
-----
Equipments Name          Total Quantity          Unit Price          Total
-----
Velvet Table Cloth          2              $8              $ 16
Microphone Set              5             $189             $ 945
-----
Grand Total: $961
Note: Fine cost will added to the grand total in case of late return

-----
Your item has been rented
Here are some available options:
-----
    Press 1 to rent an equipment
    Press 2 to return
    Press 3 to exit
-----
Please select an option: 1
                ~~~~~Equipment DETAILS~~~~~
-----
S.N.    Equipments Name          Brand          Price          Quantity
-----
1       Velvet Table Cloth          Saathi          $8              18
-----
2       Microphone Set              Audio Technica  $189             10
-----
3       Disco Light Set              Sonoff          $322             24
-----
4       7.1 Surround Sound Speaker Set  Dolby          $489              4
-----
5       Dinner Table 8x5              Panda Furnitures $344              8
-----
Please provide the SN of the item you want to rent:

```

Figure 14 Test-3 generating rent invoice with deduction of quantity of equipment

4.4 Test-4

Objective	To generate a file for returning process of equipment.
Action	2 equipment were returned.
Expected Result	Equipment should be returned and bill should be displayed with increase of quantity in the inventory.
Actual Result	Equipment was returned and bill was displayed with increase of quantity in the inventory.
Conclusion	Test was successful

Table 5 Test-4 generating file for returning equipment

```

Sikum's Equipment Rental shop

Kamalpokhari, Kathmandu | Contact No: 9855854555

-----
Equipment Details are:
-----
Name of the Costumer:SIKUM
Contact number: 0000000
Date and time of purchase: 2023-08-24 09:10:09.396576
-----

Purchase Detail are:
-----
Equipment Name      Total Quantity      Unit Price      Total
-----
Velvet Table Cloth      2              $8              $ 16
Microphone Set          5             $189            $ 945
-----
Fine: $945
Grand Total: $961

-----
Your item has been returned
Here are some available options:
-----
      Press 1 to rent an equipment
      Press 2 to return
      Press 3 to exit
-----
Please select an option: 2
                        *****Equipment DETAILS*****
-----
S.N.    Equipments Name      Brand      Price      Quantity
-----
1       Velvet Table Cloth    Saathi     $8         20
-----
2       Microphone Set        Audio Technica  $189      15
-----
3       Disco Light Set       Sonoff     $322      24
-----
4       7.1 Surround Sound Speaker Set  Dolby     $489       4
-----
5       Dinner Table 8x5      Panda Furnitures  $344       8
-----
Please provide the SN of the item you want to return:

```

Figure 15 Test-4 generating return invoice with increase of quantity of equipment

4.5 Test-5

Objective	To show the update in inventory of equipment
Action	An equipment is rented and returned.
Expected Result	The update should be shown in the inventory also in .txt file.
Actual Result	The update is shown in the inventory and also in .txt file.
Conclusion	Test was successful

Table 6 Test-5 updating inventory of equipment after renting and returning

```

Please select an option: 2
                        ~~~~~Equipment DETAILS~~~~~
-----
S.N.   Equipments Name      Brand           Price      Quantity
-----
1      Velvet Table Cloth    Saathi          $8          20
-----
2      Microphone Set        Audio Technica   $189        15
-----
3      Disco Light Set       Sonoff          $322        24
-----
4      7.1 Surround Sound Speaker Set  Dolby          $489        4
-----
5      Dinner Table 8x5      Panda Furnitures $344        8
-----
Please provide the SN of the item you want to return:

```

Figure 16 Test-5 Update in inventory

equipment				
File	Edit	View		
Velvet Table Cloth	, Saathi	, \$8	,20	
Microphone Set	, Audio Technica	, \$189	,15	
Disco Light Set	, Sonoff	, \$322	, 24	
7.1 Surround Sound Speaker Set	, Dolby	, \$489	, 4	
Dinner Table 8x5	, Panda Furnitures,	\$344	, 8	

Figure 17 Test-5 Update of inventory in .txt file

5. Conclusion

5.1 Summary

As part of the development process, I started working on building a Python-based rental management system, which represents the pinnacle of real-world application and education. Using clever file separation techniques, such as `read_write_operations.py` for data processing and `main.py` for core logic. I've created a platform that is very user-friendly. Customers are given the ability to rent items, return them, and generate bills with ease thanks to a system based on 2D lists. The experience made clear how crucial modular coding is since it ensures effective control of file operations, core activities, and user interfaces. I improved my ability to manage data structures, user interactions, and error handling throughout, understanding the IMPORTance of methodical design and documentation. The end result of this project's journey is a reliable rental management solution, which highlights my advancement in Python programming and system architecture.

5.2 limitation of the project implementation

Despite the fact that the project's implementation effectively addresses IMPORTANT features of a rental management system, there are some shortcomings and difficulties that must be acknowledged. The absence of real-time inventory updates is one drawback, as it may cause differences between the system's reported item availability and the stock on hand. Additionally, when numerous users try to rent or return products at once, the system may not handle concurrent user interactions well, potentially leading to conflicts. It could be difficult to integrate online payment gateways for bill processing because of security risks and probable technological difficulties. The project also makes the assumption that the environment is generally steady without taking into

consideration the possibility of system faults or interruptions during crucial processes. Future editions of the project might investigate real-time inventory synchronisation, develop sophisticated concurrency handling, guarantee strong payment security measures, and incorporate techniques to recover gracefully from unforeseen failures to address these practical difficulties.

5.2 Research and findings

During my research phase, I gathered IMPORTANT knowledge which had an impact on how the project was handled. Real-time inventory tracking's complexity prompted the adoption of a periodic update method, while user authentication and data security were the driving forces for the introduction of encryption and verification protocols. The need for threading and synchronisation was underlined by concurrent user interactions. The complexity of integrating safe online payments led to an emphasis on bill generation and offline payments. The research also emphasised the value of error management and system resilience to handle unforeseen problems, finally forming a practical and knowledgeable project implementation.

6. Reference

Bibliography

asp.org. (n.d.). Flowchart.
chart, I. (nov 12, 2019). *flowchart*.
educative.io. (2019). *Thonny*.
Gosling, J. (n.d.). Python. *Python.org*.
sin, a. (n.d.). *draw.io*.
Tartu, U. o. (2014 to 2018). *about.Thonny*.
techopedia. (2019). *Pseudocode*.
techopedia. (2022). *Ms word*.

Appendix

Main file

```
from datetime import datetime
from read import *
from operation import *

# Calling Function for Welcome Message
welcome_message()

loop=True
while loop==True:
    try:
        #displaying options to the user
        userinput=display_opt()

        if userinput == 1:
            rent_function()
            print("Your item has been rented")

        elif userinput == 2:
            return_function()
            print("Your item has been returned")

        elif userinput == 3:
            print("Thank you for visting us")
```

```
loop = False

elif userinput > 3 or userinput < 0:
    print("Enter the correct option")
except ValueError:
    print("Invalid Entry. Please Enter a valid numerical option")
```

read file

```
from datetime import datetime
```

```
def welcome_message():
    print("\n")
    print("-.-.-.-.-")
    print("\t\t Welcome to our system. I hope you have a good time here")
    print("-.-.-.-.-")
    print("\n")
```

```
def read_txt():
    with open("equipment.txt", 'r') as file:
        myDictionary = {}
        itemSN=1
        for line in file:
            line=line.replace("\n", "")
            myDictionary[itemSN]= line.split(",")

            itemSN= itemSN+1
        file.close()
    return myDictionary
```

```

def bill_1():
    print("-.-.-.-.-")
    print("You must enter your personal info for billing purposes. ")
    print("-.-.-.-.-")
    name = input("Please enter your Full Name: ")
    number = input("Please enter your Phone Number: ")
    return name,number

def display equipments():
    print("\t\t\t~~~~~Equipment DETAILS~~~~~\t\t\t")
    print("-.-.-.-.-")
    print("S.N. \t Equipments Name \t \t \t Brand \t \t \t Price \t \t Quantity")
    s = 1
    file = open("equipment.txt","r")
    for line in file:
        print("- - - - -")
        print(s,"\t"+line.replace(",",""))
        s+=1
    file.close()

def valid_sn1():
    print("- - - - -")
    valid_sn = int(input("Please provide the SN of the item you want to rent: "))
    print("- - - - -")
    return valid_sn

def valid_sn2():
    print("- - - - -")

```

```

valid_snn = int(input("Please provide the SN of the item you want to return: "))
print("- - - - -")
return valid_snn

```

```

def display_opt():
    print("Here are some available options: ")
    print("-----")
    print("\t Press 1 to rent an equipment")
    print("\t Press 2 to return")
    print("\t Press 3 to exit")
    print("-----")
    userinput=int(input("Please select an option: "))
    return userinput

```

Operation file

```

from datetime import datetime

```

```

from read import *

```

```

from write import *

```

```

def rent_function():
    myDictionary = read_txt()
    user_purchased equipments = []

    while True:
        display equipments()
        valid_sn = valid_sn1()
        # Valid ID
        while valid_sn <=0 or valid_sn > len(myDictionary):      #
            print("Invalid Entry.Please provide a valid equipment ID !!!")

```

```

    print("\n")
    valid_sn = int(input("Please enter S.N. of the Equipment you want to rent: "))

    print("-.....-")
    user_quantity = int(input("Please provide the number of quantity of the equipment
you want to rent: "))
    print("-.....-")
    print("-.....-")

    #Valid quantity
    get_quantity_of_selected_equipment = myDictionary[valid_sn][3]
    while user_quantity <=0 or user_quantity >
int(get_quantity_of_selected_equipment):
        print("Dear Admin, the Equipment you are looking for is not available at the
moment. Please check the table and insert the Equipmwnt")
        print("\n")

    try:
        user_quantity = int(input("Please provide a valid positive quantity: "))
    except ValueError:
        print("Please enter a valid integer value.")
        continue
    print("-.....-")

    #Update the text file

    myDictionary[valid_sn][3] = int(myDictionary[valid_sn][3]) -int(user_quantity)
    file = open("equipment.txt","w")

```

```
for values in myDictionary.values():
    file.write(str(values[0])+", "+str(values[1])+", "+str(values[2])+", "+str(values[3]))
    file.write("\n")
file.close()

# getting user purchased equipments

name_of_product = myDictionary[valid_sn][0]
quantity_selected_by_user = user_quantity
unit_price = myDictionary[valid_sn][2]
price_of_selected_equipment = myDictionary[valid_sn][2].replace("$", "")
total_price = int(price_of_selected_equipment)*int(quantity_selected_by_user)

user_purchased equipments.append([name_of_product,
quantity_selected_by_user, unit_price, total_price])

# Ask the user if they want to continue selecting
user_req = input("Dear user do you want to rent any more equipment? If yes press,
'Y' else press 'N' to exit: ").upper()

print("-.-.-.-.-")
print("\n")
print("-.-.-.-.-")

if user_req.lower()=='n':
    break
elif user_req.lower() != "y":
    continue

total = 0
```

```

for i in user_purchased equipments:
    total+=int(i[3])
grand_total = total
today_date_and_time = datetime.now()
name,phone = bill_1()
print("\n")
print("\t\t\t\t Sikum's Equipment shop")
print("\n")
print("\t\t\t\t Kamalpokhari, Kathmandu | Contact No: 9855854555")
print("\n")
print("-.-.-.-.-")
print("Equipment Details are:")
print("-.-.-.-.-")
print("Name of the Costumer:"+str(name))
print("Contact number: "+str(phone))
print("Date and time of purchase: "+str(today_date_and_time))
print("-.-.-.-.-")
print("\n")
print("Purchase Detail are:")
print("-.-.-.-.-")
print("Equipments Name \t\t Total Quantity \t\t Unit Price \t\t\tTotal")
print("-.-.-.-.-")
for i in user_purchased equipments:
    print(i[0],"\t\t",i[1],"\t\t",i[2],"\t\t","$",i[3])
print("-.-.-.-.-")

print("Grand Total: $" +str(grand_total))

```



```
print("Note: Fine cost will added to the grand total in case of late return")
```

```
write_rent(name,phone,today_date_and_time,user_purchased equipments,grand_total)
```

```
def return_function():
```

```
    myDictionary = read_txt()
```

```
    user_returned equipments = []
```

```
    fine = 0
```

```
    fineday=0
```

```
    more= True
```

```
    while more==True:
```

```
        display equipments()
```

```
        valid_snn = valid_sn2()
```

```
    # Valid ID
```

```
    while valid_snn <=0 or valid_snn > len(myDictionary):    #
```

```
        print("Invalid Entry.Please provide a valid equipment's S.N. !!!")
```

```
        print("\n")
```

```
        valid_snn = int(input("Please provide the S.N. of the equipment you want to  
return: "))
```

```
    print("-.....-")
```

```
    user_quantity = int(input("Please provide the number of quantity of the equipment  
you want to return: "))
```

```
    print("-.....-")
```

```
    print("-.....-")
```

```
    #Valid quantity
```

```

    get_quantity_of_selected_equipment = myDictionary[valid_snn][3]
    while user_quantity <=0 or user_quantity >
int(get_quantity_of_selected_equipment):
        print("Dear Admin, the equipment you are looking for is not available at the
moment. Please check the table and insert the equipment")
        print("\n")
        try:
            user_quantity = int(input("Please provide a valid positive quantity: "))
        except ValueError:
            print("Please enter a valid integer value.")
            continue
        print("-.-.-.-.-")

#Update the text file
myDictionary[valid_snn][3] = int(myDictionary[valid_snn][3])+ int(user_quantity)
file = open("equipment.txt","w")

for values in myDictionary.values():
    file.write(str(values[0])+", "+str(values[1])+", "+str(values[2])+", "+str(values[3]))
    file.write("\n")
file.close()

# getting user purchased equipments

name_of_product = myDictionary[valid_snn][0]
quantity_selected_by_user = user_quantity
unit_price = myDictionary[valid_snn][2]
price_of_selected_equipment = myDictionary[valid_snn][2].replace("$", "")
total_price = int(price_of_selected_equipment)*int(quantity_selected_by_user)

```

```
try:
    rentDays=int(input("Enter the number of days you were late: "))
    if rentDays<1:
        print("Invalid rented Days.Please Enter Again.") #print statem t
    else:
        if rentDays<=5:
            fine=0
        elif rentDays%5 !=0:
            fineday=(int(rentDays//5)+1)
            fine=int((fineday/5))*int(price_of_selected_equipment)
        else:
            fine=(rentDays-5)*int(price_of_selected_equipment)

    anss= input("Dear user do you want to return any more equipment? If yes
press, 'Y' else press 'Enter' key.").upper()

    if anss=="Y":
        print("\n")
        more=True
    else:
        more=False

    grand_total=0
    if anss=="Y":
        more=True
    else:
        total=0
        for i in user_returned equipments:
            total=total+int(i[3])
```

```

        grand_total= total+fine
        today_date_and_time = datetime.now()
        #break
    except ValueError:
        print("Invalid days.Please try Again")
        print("\n")

    user_returned equipments.append([name_of_product, quantity_selected_by_user,
unit_price,total_price ,rentDays,fine])

    # Ask the user if they want to continue selecting
    print("-.-.-.-.-")
    print("\n")

total = 0
for i in user_returned equipments:
    total+=int(i[3])
grand_total = total
today_date_and_time = datetime.now()
name,phone = bill_1()
print("\n")
print("\t\t\t\t Sikum's Equipment Rental shop")
print("\n")
print("\t\t\t\t Kamalpokhari, Kathmandu | Contact No: 9855854555")
print("\n")
print("-.-.-.-.-")
print("Equipment Details are:")
print("-.-.-.-.-")
print("Name of the Costumer:"+str(name))

```

```

print("Contact number: "+str(phone))
print("Date and time of purchase: "+str(today_date_and_time))
print("-.....-")
print("\n")
print("Purchase Detail are:")
print("-.....-")
print("Equioment Name \t\t Total Quantity \t\t Unit Price \t\t\tTotal")
print("-.....-")
for i in user_returned equipments:
    print(i[0], "\t\t", i[1], "\t\t", i[2], "\t\t", "$", i[3])
print("-.....-")
print("Fine: $" + str(fine))
print("Grand Total: $" + str(grand_total))

```

```

write_return(name, phone, today_date_and_time, user_returned equipments, grand_total)

```

Write file

```

from operation import *

```

```

def
write_rent(name, phone, today_date_and_time, user_purchased equipments, grand_total)
:
    print("\n")
    print("-----")
)
with open(name+phone+".txt", "w") as file:
    file.write("\n")
    file.write("\t\t\t\t\t Sikum's Equipment Rental shop")
    file.write("\n")

```

```
file.write("\t\t\t\t Kamalpokhari, Kathmandu | Contact No: 9855854555")
file.write("\n")
file.write("-----")
file.write("Equipment Details are:")
file.write("-----")
file.write("Name of the Costumer:"+str(name) +"\n")
file.write("Contact number: "+str(phone)+"\n")
file.write("Date and time of purchase: "+str(today_date_and_time)+"\n")
file.write("-----")
file.write("\n")
file.write("Purchase Detail are:")
file.write("-----")
file.write("Equioment Name \t\t Total Quantity \t\t Unit Price \t\tTotal")
file.write("-----")
for i in user_purchased equipments:
    file.write(str(i[0])+"\t\t"+str(i[1])+"\t\t\t"+str(i[2])+"\t\t"+"$"+str(i[3])+"\n")
    file.write("-----")
file.write("Grand Total: $" +str(grand_total))
file.write("Note: Fine cost will added to the grand total in case of delay")

def
write_return(name,phone,today_date_and_time,user_returned equipments,grand_total)
:
print("\n")
print("-----")
with open(name+phone+".txt","w") as file:
```

```

file.write("\n")
file.write("\t\t\t\t Sikum's Equipment Rental shop")
file.write("\n")
file.write("\t\t\t\t Kamalpokhari, Kathmandu | Contact No: 9855854555")
file.write("\n")
file.write("-----")
-----")
file.write("Equipment Details are:")
file.write("-----")
file.write("Name of the Costumer:"+str(name)+"\n")
file.write("Contact number: "+str(phone)+"\n")
file.write("Date and time of purchase: "+str(today_date_and_time)+"\n")
file.write("-----")
-----")
file.write("\n")
file.write("Purchase Detail are:")
file.write("-----")
file.write("Equipments Name \t\t Total Quantity \t\t Unit Price \t\t\tTotal")
file.write("-----")
-----")
for i in user_returned equipments:
    file.write(str(i[0])+"\t\t"+str(i[1])+"\t\t\t"+str(i[2])+"\t\t"+"$"+str(i[3])+"\n")
    file.write("-----")
-----")

#file.write("Fine: $" +str(fine))
file.write("Grant Total: $" +str(grand_total))

```