LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Rupesh Kumarmahato

London Met ID: 23047358

College ID: NP01NT4A230053

Assignment Due Date: Tuesday, May 7, 2024

Assignment Submission Date: Tuesday, May 7, 2024

Word Count: 2402

Project File Links:

YouTube Link:	Keep Unlisted YouTube URL of your Project Here
Google Drive Link:	**https://drive.google.com/drive/folders/1iWNKlEc9Mf2EBnNhwVoenmqC1CUYdLXu?usp=drive_link**

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.

Acknowledgement

I want to express my heartfelt gratitude to all those who played a part in the completion of this report. I extend my deepest appreciation to Mr. Prayag Koirala and Mr. Monil Adhikari for their invaluable assistance, guidance, and unwavering support throughout the entire process.

A special mention goes to Mrs. Shresha Raj Bhandarifor her insightful feedback and encouragement, which greatly enhanced the quality of this work.Furthermore, I would like to acknowledge Islington College for generously providing resources and facilities essential for conducting the research. This report would not have come to fruition without the contributions of each and every individual mentioned above. Thank you from the bottom of my heart

Abstract

This document outlines a program developed for the Fundamentals of Computing module's coursework using IDLE. It is structured into different sections, each focusing on various aspects of the project. It starts with the goals and objectives of the coursework, followed by a discussion on the tools used and their roles in the development process. Algorithms and pseudocode are included to demonstrate real-world applications, complemented by flowcharts to aid in understanding the program's logic and practical uses. Data structures and the testing process are also covered, detailing various methods used to detect and address errors. Additionally, screenshots of the program are provided to visually demonstrate its functionality. The report concludes with a project summary, objectives, outcomes, and a list of references and citations.

Contents

Table of figures

Table of tables

1.Introduction

Python is widely utilized for creating websites and software, automating tasks, analyzing data, and visualizing data. Its simplicity has led to its adoption by numerous non-programmers, including accountants and scientists, who use it for various everyday activities such as managing finances.courseera. It supports various programming paradigms, including object-oriented, functional, and procedural styles. In this programming language indentation matters rather than brackets.

1.2. About the project

This is an individual coursework assigned in CS4051NI/CC4059NI module. This coursework covers 60% weightage of this module. This course work is mainly based on the basic concept of Python. This course work is to develop a Land Rental System where customer can borrow land on rent which are available. the record of land are maintained in th text file which is updated after getting land borrowed or returned and if customers do not return land on time than fine is also imposed on them which is added to total amount in invoice.

1.3. Aim and objective

As the course is about to develop a Land Rental System its aim is to make user friendly system where customer can easily borrow land which are available and to provide reliable service to them. Also, its aim is to keep the record of every land which is rented or returned and make available accordingly.

The main objective of this course work are as follow.

- Display Available Lands: Read and show available lands for rent.
- Handle Transactions: Manage renting and returning of lands, updating availability status.

- Generate Notes/Invoices: Create .txt files for transactions, including details like kitta number, city, direction, area, customer info, dates, duration, and total amount.
- Calculate Cumulative Rent: Sum up rents if customer rents multiple lands.
- Late Fee Handling: Apply monthly fines for late returns, updating the file accordingly.

1.4. Tools used

1. IDLE: IDLE, Python's Integrated Development and Learning Environment, enables easy Python coding. Similar to Python Shell, it executes single statements and creates, edits, and runs Python scripts. IDLE includes a comprehensive text editor with features such as syntax highlighting, autocompletion, and smart indenting. It also features a debugger with stepping and breakpoints for easier debugging. (Python docs, n.d.)

2. MS word: Microsoft Word is a leading word processing software known for its user-friendly interface, formatting options, and collaboration features. It's part of the Microsoft Office suite and offers templates, cloud integration, and compatibility with various file formats. Word is widely used for creating documents ranging from simple letters to complex reports, both professionally and personally.

3. draw.io : Draw.io is a free web-based tool for creating various types of diagrams such as flowcharts, diagrams, mind maps, and organization charts. It's seamlessly integrated with Google Drive, enabling automatic saving of your work in your Google Workspace or Gmail account. (Paraschiv, 2023)

2. Discussion and analysis

2.1 Algorithm

An algorithm consists of a series of instructions that a computer follows to perform calculations or solve problems. Formally, it's a finite set of specific instructions executed in a particular sequence to achieve a specific task. It represents the logic to solve a problem, rather than the entire program or code. (Upadhyay, 2023)

Step 1: Start

Step 2: Call function main

Step 3: Display choice to the user

Step 4: Take user input for choice

If user input is equal to 1 go to next step

If user input is equal to 2 go to step 24

If user input is equal to 3 go to step 37

Step 5: then call function display and rent_land

Step 6: Take user input for name and store it in customer_name

Step 7: Take user iput for duration and store it in rent_duration

Step 8: If it is valid go to step 10

Step 9: If not display Error: Invalid input. Please enter a valid number and go to step3

Step 10: initialize total_rent_amount to 0

Step 11: call the function read_land_info and store it in land_info

step 12: Create an empty list rented_lands to store the information of lands that the customer has rented.

step 13: Start an indefinite loop with while True to allow the user to rent multiple lands until they finish.

Step 14: Take the user input for the kitta number of the land they want to rent, or type 'done' to finish. Store the input in kitta_number.

Step 15: check if the input is done breakout of the loop

Step 16: Iterate through each land in land_info

Step 17: Check if the kitta number matches and if the land is available go to step 20

Step 18: If the land is not available

Step 19: Print a message indicating that the land with the given kitta number is not available for rent.

Step 20: Mark the land as rented by changing its status to 'Not Available'.

Get the current date and time.

Calculate the rent amount by multiplying the land's rate with the rent duration.

Add the rent amount to total_rent_amount.

Append the rented land information to rented_lands.

Print a success message with the rented land's kitta number and rent amount.

Step 21: Check if there are any lands rented (if rented_lands is not empty):

Prepare a rental note containing details of the rented lands, customer's name, rent date, rent duration, and total rent amount.

Open a file in write mode with the name "Rent_Invoice_customer_name.txt" and write the rental note into it.

Print a success message indicating that the invoice is generated successfully.

Call the function write_land_info() to update the land information after renting.

step 22: handle the error

step23: End the try block

step 24: call function display and return_land

step 25: start try block

step 26: Take user input kitta number and customer name

step 27: get current date and time

step 28:input rent duration

step 29: read land information by calling read_land_info function

step 30: check the kitta number and availability by iterating through each land information.

if kitta number matches and land is available then go to next step if not print land was not rented or the Kitta Number doesn't exist.

step31: calculate rent amount and fine

rent amount = monthly rent of land * land duration

if the rent duration > 12 then total amount = fine_rate * overdue_months + rent amount

step 32:create fine invoice

if there is a fine create a fine invoice text file including details such as Kitta Number, City/District, Land Faced, fine rate, overdue months, and fine amount.

step 33: create return invoice

create invoice text file including details such as Customer Name, Kitta Number, City/District, Return Date, Rent Duration, Area, Total Amount, and Fine Amount.

step 34: print land has been returned successfully along with the total amount.

step 35: update land information by calling function write_land_info

step 36: handle invalid input

step 37: Exit

step 38: End

2.2. Pseudocode

In data science or web development, pseudocode serves as a technique for describing the distinct steps of an algorithm in a manner that's easily understandable for individuals with basic programming knowledge, aiding in clarity and comprehension. (Urwin, 2024)

Main Module

```
DECLARE FUNCTION display():
        OPEN "land_info.txt" in read mode as file
        READ all lines from file into lines
        INITIALIZE data list

        FOR EACH in lines:
                split line by comma
                append split parts to data list

        PRINT header
        PRINT line of dashes
        END FOR

        FOR EACH item in data:
                PRINT item details with appropriate spacing
        END FOR
END FUNCTION

DECLARE FUNCTION main():
        PRINT welcome message
        PRINT options for user
```

```
    DO WHILE True:
        PRINT menu options
        GET user choice

        IF choice is '1':
            DISPLAY land information
            CALL rent_land()
        ELIF choice is '2':
            DISPLAYland information
            CALL return_land()
        ELIF choice is '3':
            PRINT thank you message
            BREAK
        ELIF
            PRINT error message for invalid choice
        ENDIF
    END DO WHILE
END FUNCTION


IF this module is run directly:
    CALL main()
ENDIF
```

Operation Module

```
IMPORT datetime
FROM read IMPORT read_land_info
FROM write MPORT write_land_info

DECLARE FUNCTION rent_land
```

```
    TRY
        GET customer_name
        GET rent_duration
        SET total_rent_amount = 0
        SET land_info = CALL read_land_info
        INITIALIZE rented_lands

        DO WHILE TRUE
            GET kitta_number
            IF kitta_number is 'done':
            Exit loop
            found = False
            FOR EACH land in land_info
                IF land's kitta_number matches input kitta_number and land
                is 'Available':
                SET land status to 'Not Available'
                GET current date and time
                CALCULATE rent_date
                CALCULATE rent_amount
                ADD rent_amount to total_rent_amount
                APPEND land to rented_lands
                DISPLAY success message
                SET found to True
                BREAK
            END FOR
            IF not found:
                DISPLAY land not available message
            END IF

IF rented_lands is not empty:
  CREATE note
```

```
        FOR EACH land in rented_lands:
                APPEND land details to note
                APPEND customer details and rent information to note
                WRITE note to file
                DISPLAY invoice generated message
                WRITE updated land_info to file
        ELSE
                DISPLAY no lands rented message
        END IF
    EXCEPT ValueError:
        DISPLAY invalid input message


DECLARE FUNCTION return_land
        TRY
                GET kitta_number
                GET customer_name
                GET current date and time
                CALCULATE return_date
                GET rent_duration
                SET land_info =  CALL read_land_info

                FOR EACH in land_info:
                        IF land's kitta_number matches input kitta_number and land is 'Not
        Available'
                                SET land status to 'Available'
                                CALCULATE rent_amount
                                CALCULATE return_amount
                                CALCULATE overdue_months
                                CALCULATE fine
                                ADD fine to return_amount
```

```
                    IF fine is greater than 0:
                        CREATE fine_note
                        WRITE fine_note to file
                        CREATE return_note
                        WRITE return_note to file
                    DISPLAY success message
                    WRITE updated land_info to file

                DISPLAY land not rented or kitta number does not exist message
            EXCEPT ValueError:
                DISPLAY invalid input message
END FUNCTION
```

Read Module

```
DECLARE FUNCTION read_land_info
        INITIALIZE land_info
        OPEN "land_info.txt" file for reading
        FOR EACH line in the file
                SPLIT the line by commas
                APPEND the resulting list to land_info
        END FOR
        CLOSE the file
        RETURN land_info
END FUNCTION
```

Write Module

```
DECLARE FUNCTION write_land_info
```

```
    OPEN "land_info.txt" in write mode AS file
    FOR EACH land IN land_info DO
        WRITE (JOIN elements of land WITH ', ' AND ADD a newline) TO file
    END FOR
    CLOSE file
END FUNCTION
```

2.3.Data structure

Data structures are essential tools for organizing data efficiently in programming, forming the foundation of software development. Python, compared to other languages, provides a simpler approach to learning these fundamental concepts. Python offers various built-in data structures such as lists, tuples, dictionaries, alongside more advanced ones like trees and graphs. But in this project I have used lists.

- Lists

  Python list is similar to arrays in other languages, are ordered collections of data that offer great flexibility, as the items within a list can be of different types. The implementation of Python lists resembles vectors in C++ or ArrayLists in Java. However, operations like inserting or deleting elements at the beginning of the list can be costly because all elements need to be shifted. Similarly, if preallocated memory becomes full, insertion and deletion at the end of the list can also become expensive. (GeekForGeeks, 2023)

3.program

3.1.implementation of program

This program is used to rent land from Techno Property Nepal. The programs asks user to enter their choice and according to which the user user are able to rent or return land. Bill is also generated at the time of renting or returning land and this programs runs in a loop until the user exit.

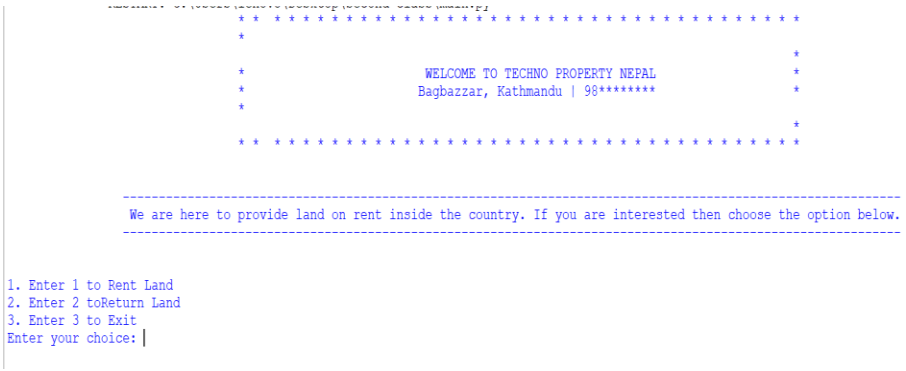

Figure 1: Implementation of program

3.2 Renting land

If the users choice is 1 then the code works like this.

```
Enter your choice: 1
Kitta No.   Location    Direction   Anna       Price      Availability
-------------------------------------------------------------------
110         Kathmandu     North       4          50000      Available
120         Pokhara       East        5          60000      Not Available
130         Lalitpur      South       10         100000     Not Available
140         Kathmandu     North       4          50000      Not Available
150         Pokhara       East        5          60000      Not Available
160         Lalitpur      South       10         100000     Not Available
Enter your name: rupesh
Enter the duration of rent (in months): 5
Enter Kitta Number of the land you want to rent (or type 'done' to finish): 110
Enter the number of annas you want to rent: 4
Land with Kitta Number 110 rented successfully. Rent amount: Rs. 250000
Enter Kitta Number of the land you want to rent (or type 'done' to finish): done
Invoice generated successfully for rupesh . Total rent amount: Rs. 250000
```

Figure 2:Screenshots of when user enter number 1 for renting land

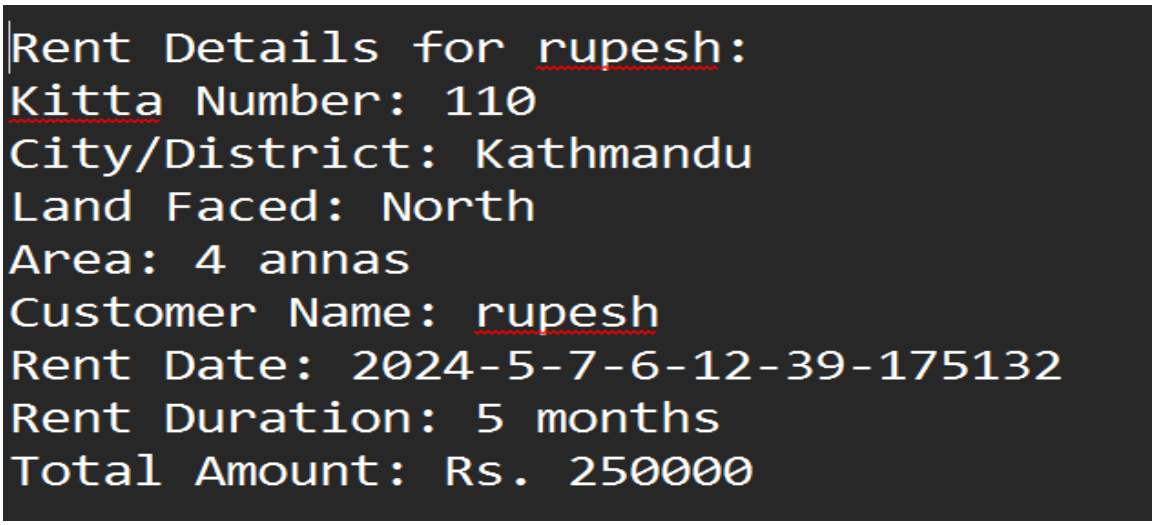

Figure 3: Screenshot of invoice generated after renting land

3.3 Creation of txt file

Before creating

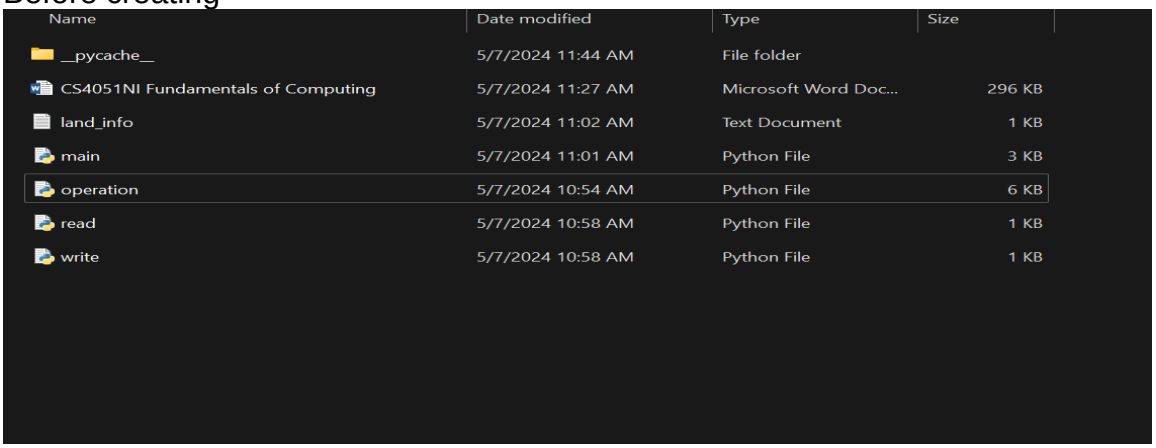

Figure 4: Screenshot before creation of text file

After creating txt file

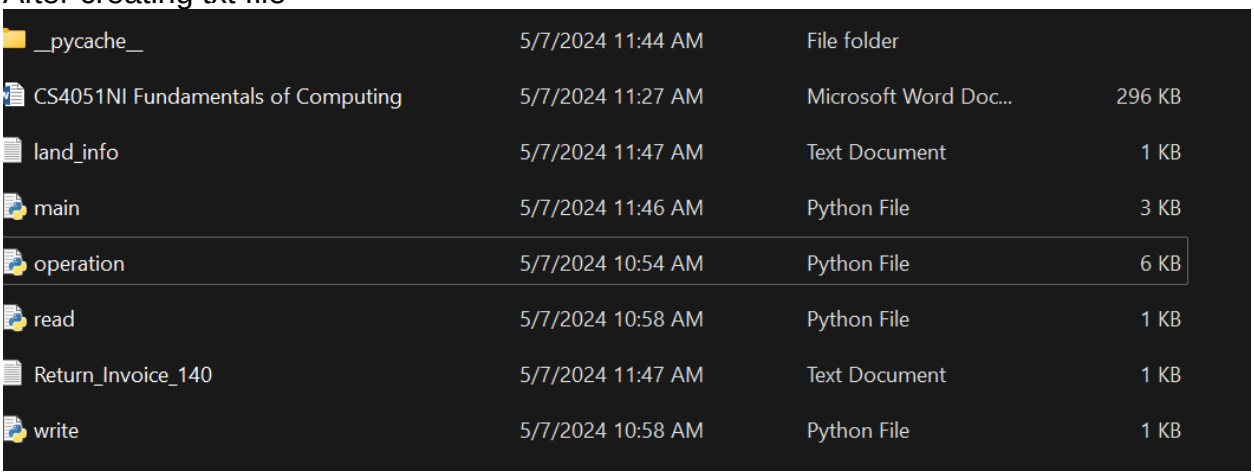

Figure 5: Screenshot after creating text file

3.4 opening text file and showing the bill

After returning land invoice is generated as text file which includes details of land and customer.

```
Return Details:
Customer Name: fghjk
Kitta Number: 140
City/District: Kathmandu
Land Faced: North
Return Date: 2024-5-7-11-47-13-49784
Rent Duration: 4 months
Area: 4
Total Amount: Rs. 200000
Fine Amount: Rs. 0
```

Figure 6: Screenshot of displaying details of bill

3.5 End of Program

If the users choice is 3 then the program ends displaying thankyou message

```
1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: 3
Thank You For Visiting Us!!
```

Figure 7: Screenshot of program when user input 3

4.Testing

Test 1:

Test	1
Action performed	When the user is asked to enter number of month and user enter it in word rather than in number.
Expected result	The program should throw error displaying error message to input valid number.
Actual result	The program throws error displaying error message.
conclusion	Hence the test was successful

Table 1: Test 1

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                               *
*                        WELCOME TO TECHNO PROPERTY NEPAL                       *
*                      Bagbazzar, Kathmandu | 98********                        *
*                                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

----------------------------------------------------------------------------------------------
 We are here to provide land on rent inside the country. If you are interested then choose the option below.
----------------------------------------------------------------------------------------------

1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: 1
Kitta No.   Location    Direction   Anna      Price     Availability
----------------------------------------------------------------------
110         Kathmandu    North       4         50000     Not Available
120         Pokhara      East        5         60000     Not Available
130         Lalitpur     South       10        100000    Not Available
140         Kathmandu    North       4         50000     Not Available
150         Pokhara      East        5         60000     Not Available
160         Lalitpur     South       10        100000    Not Available
Enter your name: ghj
Enter the duration of rent (in months): six
Error: Invalid input. Please enter a valid number.

1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: |
```

Figure 8: Screenshot when user enter invalid value

Test 3

Test	3
Action performed	User chose 1 to rent land and rented two land of kitta number 110 and 140 by providing required details.
Expected result	Single invoice should be generated for both lands.
Actual result	Single invoice was generated for both lands.
Conclusion	Hence, the test was successful.

Table 2: test 3

```
1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: 1
Kitta No.   Location    Direction   Anna      Price     Availability
----------------------------------------------------------------------
110         Kathmandu    North       4         50000     Available
120         Pokhara      East        5         60000     Not Available
130         Lalitpur     South       10        100000    Not Available
140         Kathmandu    North       4         50000     Available
150         Pokhara      East        5         60000     Available
160         Lalitpur     South       10        100000    Not Available
Enter your name: fgh
Enter the duration of rent (in months): 4
Enter Kitta Number of the land you want to rent (or type 'done' to finish): 110
Enter the number of annas you want to rent: 4
Land with Kitta Number 110 rented successfully. Rent amount: Rs. 200000
Enter Kitta Number of the land you want to rent (or type 'done' to finish): 140
Enter the number of annas you want to rent: 4
Land with Kitta Number 140 rented successfully. Rent amount: Rs. 200000
Enter Kitta Number of the land you want to rent (or type 'done' to finish): done
Invoice generated successfully for fgh . Total rent amount: Rs. 400000
```

Figure 9: Renting two land at a time

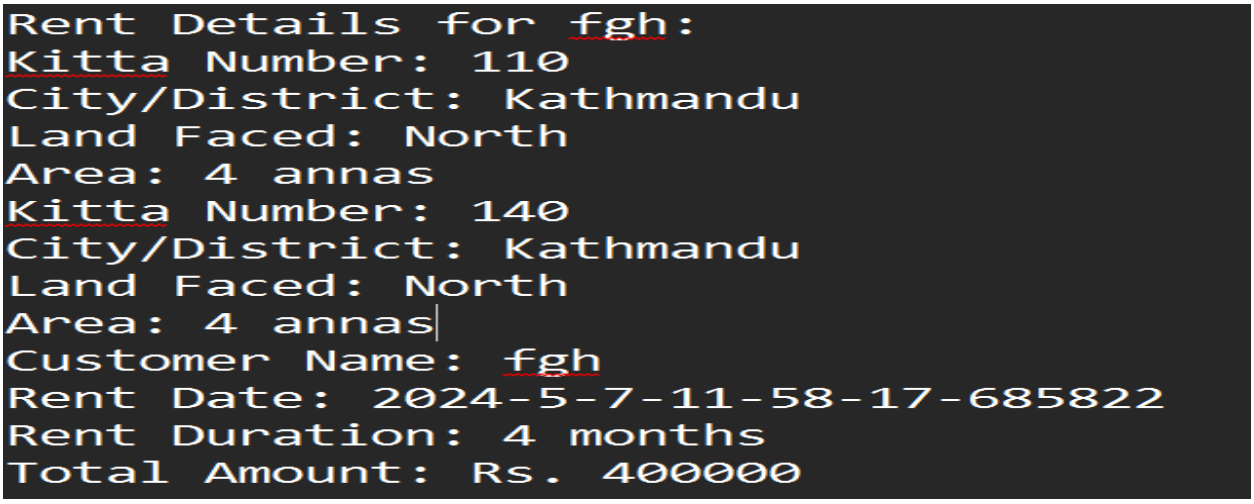

Figure 10: single invoice forboth land

Test 5

Test	2
Action performed	User chose the option 2 and returned land of kitta number 110.
Expected result	When the land is returned it should be updated and availability of land of kitta number 110 should be updated to Available.
Actual result	When the land of kitta number 110 is returned its availability is updated to Available.
conclusion	Hence, the result is successful.

Table 3: Test 5

```
------------------------------------------------------------------------------------------------------------
 We are here to provide land on rent inside the country. If you are interested then choose the option below.
------------------------------------------------------------------------------------------------------------

1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: 2
Kitta No.   Location    Direction   Anna       Price      Availability
-------------------------------------------------------------------
110         Kathmandu     North       4          50000      Not Available
120         Pokhara       East        5          60000      Not Available
130         Lalitpur      South       10         100000     Not Available
140         Kathmandu     North       4          50000      Not Available
150         Pokhara       East        5          60000      Not Available
160         Lalitpur      South       10         100000     Not Available
Enter Kitta Number of the land you are returning: 110
Enter your name: rupesh
Enter the duration of rent (in months): 5
Land returned successfully. Total Amount: Rs. 250000.

1. Enter 1 to Rent Land
2. Enter 2 toReturn Land
3. Enter 3 to Exit
Enter your choice: 1
Kitta No.   Location    Direction   Anna       Price      Availability
-------------------------------------------------------------------
110         Kathmandu     North       4          50000      Available
120         Pokhara       East        5          60000      Not Available
130         Lalitpur      South       10         100000     Not Available
140         Kathmandu     North       4          50000      Not Available
150         Pokhara       East        5          60000      Not Available
160         Lalitpur      South       10         100000     Not Available
```

Figure 11: Screenshot of updated file after returning land

5. Conclusion

This Python programming course was undertaken individually, focusing on researching and improving skills with various tools. Tools such as IDLE, Microsoft Word, and draw.io were utilized. The coding aspect was completed in IDLE, while documentation was handled in Microsoft Word, and draw.io was used to create flowcharts. Throughout the course, I encountered some difficulties and sought assistance from my tutor as well as various online resources to find solutions. This project helped me develop the ability to work independently. The report provides a detailed overview of the project, including its creation process, instructions for usage, testing results, and algorithms and flowcharts to aid in understanding the logic.

The project was organized to make it easier to reuse code and break down complex problems into smaller, more understandable pieces. We focused on accessing and

changing data from a specific module, "CS4051NI/CC4059NI Fundamental of Computing", in a functional way. We also made sure the program could handle any problems that might come up. The main aim of the project was to help understand how data structures work and how they help organize information efficiently.

I had an awesome experience while doing this coursework. I learned to do basic python programming which I might be using in my working career as well. This coursework also developed my research skill. And now I think have sufficient knowledge of basic Python programming after doing this course work.

Bibliography

GeekForGeeks, 2023. *GeekForGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/python-data-structures/
[Accessed 4 May 2024].

Paraschiv, L., 2023. *FOTC.* [Online]
Available at: https://fotc.com/blog/draw-io-online-guide/
[Accessed 4 May 2024].

Python docs, n.d. *Python docs.* [Online]
Available at: https://docs.python.org/3/library/idle.html
[Accessed 5 May 2024].

Upadhyay, S., 2023. *simplilearn.com.* [Online]
Available at: https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm
[Accessed 5 May 2024].

Urwin, M., 2024. *Built In.* [Online]
Available at: https://builtin.com/data-science/pseudocode
[Accessed 5 May 2024].

Appendix

read module

```
def read_land_info():
    land_info = []
    with open("land_info.txt", 'r') as file:
        for line in file:
            land_info.append(line.strip().split(', '))
    return land_info
```

write module

```
def write_land_info(land_info):
    with open("land_info.txt", 'w') as file:
        for land in land_info:
            file.write(', '.join(land) + '\n')
```

operation module

```
import datetime
from read import read_land_info
from write import write_land_info



def rent_land():
    try:
        customer_name = input("Enter your name: ")
        rent_duration = int(input("Enter the duration of rent (in months): "))
        total_rent_amount = 0

        land_info = read_land_info()
        rented_lands = []
```

```
        while True:
            kitta_number = input("Enter Kitta Number of the land you want to rent (or type
'done' to finish): ")
            if kitta_number.lower() == 'done':
                break
            anna_demand = int(input("Enter the number of annas you want to rent: "))

            found = False
            for land in land_info:
                if land[0] == kitta_number and land[5] == 'Available':
                    if int(land[3]) != anna_demand:
                        print("Invalid input. Not enough annas available for rent.")
                        return
                    land[5] = 'Not Available'
                    now = datetime.datetime.now()
                    rent_date = "-".join([str(now.year), str(now.month), str(now.day),
str(now.hour), str(now.minute), str(now.second), str(now.microsecond)])
                    rent_amount = int(land[4]) * rent_duration
                    total_rent_amount += rent_amount
                    rented_lands.append(land)
                    print("Land with Kitta Number", kitta_number, "rented successfully. Rent
amount: Rs.", rent_amount)
                    found = True
                    break
            if not found:
                print("Land with Kitta Number", kitta_number, "not available for rent.")

        if rented_lands:
            note = "Rent Details for " + customer_name + ":\n"
            for land in rented_lands:
```

```
            note += "Kitta Number: " + land[0] + "\nCity/District: " + land[1] + "\nLand Faced: " + land[2] + "\nArea: " + land[3] + " annas\n"
            note += "Customer Name: " + customer_name + "\nRent Date: " + rent_date + "\nRent Duration: " + str(rent_duration) + " months\nTotal Amount: Rs. " + str(total_rent_amount) + "\n"

            with open("Rent_Invoice_" + customer_name + ".txt", 'w') as invoice_file:
                invoice_file.write(note)
                print("Invoice generated successfully for", customer_name, ". Total rent amount: Rs.", total_rent_amount)

            write_land_info(land_info)
        else:
            print("No lands rented.")

    except ValueError:
        print("Error: Invalid input. Please enter a valid number.")

def return_land():
    try:
        kitta_number = input("Enter Kitta Number of the land you are returning: ")
        customer_name = input("Enter your name: ")
        now = datetime.datetime.now()
        return_date = "-".join([str(now.year), str(now.month), str(now.day), str(now.hour), str(now.minute), str(now.second), str(now.microsecond)])
        rent_duration = int(input("Enter the duration of rent (in months): "))

        land_info = read_land_info()
        for land in land_info:
            if land[0] == kitta_number and land[5] == 'Not Available':
                land[5] = 'Available'
```

```
            rent_amount = int(land[4]) * rent_duration
            return_amount = rent_amount

            # Calculate fine for overdue months
            overdue_months = rent_duration - 12 if rent_duration > 12 else 0
            fine_rate = 10000  # Fine rate per month
            fine = fine_rate * overdue_months
            return_amount += fine  # Add fine to return amount

            # Create fine invoice
            if fine > 0:
                fine_note = "Fine Applied:\nKitta Number: " + land[0] + "\nCity/District: " +
land[1] + "\nLand Faced: " + land[2] + "\nFine Rate: Rs. " + str(fine_rate) +
"/month\nOverdue Months: " + str(overdue_months) + "\nFine Amount: Rs. " + str(fine) +
"\n"
                with open("Fine_Invoice_" + kitta_number + ".txt", 'w') as invoice_file:
                    invoice_file.write(fine_note)

            # Create return invoice
            return_note = "Return Details:\nCustomer Name: " + customer_name +
"\nKitta Number: " + land[0] + "\nCity/District: " + land[1] + "\nLand Faced: " + land[2] +
"\nReturn Date: " + return_date + "\nRent Duration: " + str(rent_duration) + "
months\nArea: " + land[3] + "\nTotal Amount: Rs. " + str(return_amount) + "\nFine
Amount: Rs. " + str(fine) + "\n"
            with open("Return_Invoice_" + land[0] + ".txt", 'w') as invoice_file:
                invoice_file.write(return_note)

            print("Land returned successfully. Total Amount: Rs. " + str(return_amount) +
".")
            write_land_info(land_info)
            return
```

```
        print("Land was not rented or kitta number does not exist.")

    except ValueError:
        print("Error: Invalid input. Please enter a valid number.")
```

main module

```
from operation import rent_land, return_land
def display():

    with open("land_info.txt", 'r') as file:
        lines = file.readlines()
        data = []
        for line in lines:
            parts = line.strip().split(',')
            data.append(parts)
        print("Kitta No.   Location    Direction   Anna      Price      Availability")
        print("-" * 70)


        for item in data:
            print(
                item[0], " " * (8 - len(str(item[0]))),
                item[1], " " * (12 - len(item[1])),
                item[2], " " * (10 - len(item[2])),
                item[3], " " * (8 - len(str(item[3]))),
                item[4], " " * (9 - len(str(item[4]))),
                item[5], " " * (14 - len(item[5]))
            )
                #print(line.strip())
```

```
def main():
    print(" \t\t\t\t* *  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ")
    print(" \t\t\t\t* \t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t     *")
    print(" \t\t\t\t* " + "\t\t\t  WELCOME TO TECHNO PROPERTY NEPAL" +"\t\t     *")
    print(" \t\t\t\t* " + "\t\t\t Bagbazzar, Kathmandu | 98********" +"\t\t     *")
    print(" \t\t\t\t* \t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t     *")
    print(" \t\t\t\t* *  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * \n\n")
    print(" \t\t-----------------------------------------------------------------------------------------------
-------")
    print("\t\t We are here to provide land on rent inside the country. If you are interested
then choose the option below.")
    print(" \t\t-----------------------------------------------------------------------------------------------
-------\n")


    while True:
        print("\n1. Enter 1 to Rent Land\n2. Enter 2 toReturn Land\n3. Enter 3 to Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            display()
            rent_land()
        elif choice == '2':
            display()
            return_land()
        elif choice == '3':
            print("Thank You For Visiting Us!!")
            break
        else:
            print("Invalid choice. Please enter a valid option.")
```

```
if __name__ == "__main__":
    main()
```