1st Sit COURSEWORK Question Paper: Spring Semester 2024

Module Code:	**CS4051NI/CC4059NI**
Module Title:	**Fundamentals of Computing**
Module Leader:	**Mr. Hrishav Tandukar** (Islington College)

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **60%** of your total module grades.
Submission Date:	**Tuesday, 7 May 2024, before 01:00 PM**
When Coursework is given out:	**Week 20**
Submission Instructions:	Submit the following to Islington College's MST Portal before the due date: ● **Soft copy of the report** ● **Zip file with source code of the program**
Warning:	London Metropolitan University and Islington College takes Plagiarism seriously. Offenders will be dealt with sternly.

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed overleaf. Please sign below to say that you have read and understand these extracts:

Extracts from University *Regulations on Cheating, Plagiarism and Collusion*

Section 2.3: *"The following broad types of offence can be identified and are provided as indicative examples ….*

(i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii) Falsifying data in experimental results.

(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v) Collusion to present joint work as the work solely of one individual.

(vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation, and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "$e = mc^2$ (Einstein 1905)". A *reference* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

School of Computing, FLSC

This module is assessed by coursework (60%). For the coursework, the students are required to develop an application based on detailed guidance on given specifications. Through the coursework students should be able to:

- ✓ Develop a ***Land Rental System***
- ✓ Describe the program
- ✓ Test the program with some sample data to demonstrate its behaviour
- ✓ Write a report to present the work

- Students are required to submit weekly assignment along with related research evidence

- The guideline for the Documentation/Development part is given below:

    - Components to be included are: cover page, and table of contents, figures and tables, footer.

    - The report must have an **Introduction** section with definite goals and objectives (approx. 400 words or more).

    - The report must have a **Discussion and Analysis** section in which students need to explain how the program was developed.

1. Scenario

TechnoPropertyNepal is a private company that has a stock of different land on a contract basis in different locations in Nepal. Thus, the company allows its clients to rent the available land in various locations. The land stocks which are available are to be charged on a monthly basis.

The TechnoPropertyNepal manages the availability of the land in a single text file. The program must be able to read the text file and display the lands that are available or not available for rent. Adding to that, the program should carry out the changes according to the nature of the transaction (i.e., renting of the land and returning the land to the company after the termination of the contract). With each transaction made, a note or invoice must be generated. For instance, if the customer rents land for 6 months, then the status in the main text file must be modified to not available. Similarly, if the customer returns the land after the termination of the rental contract, the Status must be updated to available.

A sample format of the text file including the information about the Land Renting System is as follows:

101, Kathmandu, North, 4, 50000, Available

102, Pokhara, East, 5, 60000, Not Available

103, Lalitpur, South, 10, 100000, Available

**1st column shows the kitta number of lands, 2nd column shows the name of the city/district, 3rd column shows the direction of the land (Land Faced), 4th column shows anna, 5th column shows the price in Nepalese Rupee, 6th column shows the availability status of the land.*

Note: You can use your own format and add other information too.

A note/invoice should be generated for each transaction. When land is rented, a note/invoice should be generated (as a .txt file) which must contain the kitta number of lands, name of the city/district, the direction of the land (Land Faced), area of land (anna), name of the customer, date and time of rent, the duration of rent, and the total amount. Also, if a customer decides to rent more than one land, then the amount should be added up for all the rented lands.

When the land is returned back, a note/invoice should be generated again which should include the name of the customer, kitta number of the land, name of the city/district, the direction of the land (Land Faced), date and time of returning, the duration of rent, area of land(anna), and the total amount. However, if a client is unable to renew the contract on time and is late to return the land, in such case a fine should be applied on a monthly basis which should be written to the file again.

** The format of the notes/invoices is up to you. But each file should have a unique name. The customer has to take the available land as a whole. For example, if the kitta number is 101, it includes an area comprising 4 annas of land. The customer must rent the entire 4 annas of land; they cannot rent less than that.*

2. **Algorithm**

- An algorithm should be developed for the application where everything the program does should be taken into account. The algorithm should be described in steps, pseudocode, and flowcharts should also be included.

3. **Data Structures**

- The programming should be done using data structures and operations in Python for input/output, character and string processing, and data storage.
- It can use any primitive or complex data structures which might be necessary for holding the data (pairs, lists, strings, dictionaries, etc.)
- The choice of data structures must be specified in the report.

4. **Program**

- The program must work in a loop, displaying the available lands and waiting for the administrator to enter details of the customers. The program should not close unless the administrator decides to do so.
- The program must check the input data, displaying error messages whenever unwanted data is entered, for example if some string value is entered where a numerical value is expected.
- The program must be implemented in a modular way with separate functions for the different operations such as input/output, reading files, generating invoices/notes, etc.

5. **Description**

- The program must be described in terms of its structure and behavior.
- It can be presented using text and structural charts, flowcharts or other diagrams as needed.
    - The report must have a **Testing** section with evidence (through appropriate screenshots) that has been carried out for the program.
    - The report must include a **Conclusion** section where they need to present their findings of the development and research (approx. 300 words).
- Demonstration of the project.
    - If any individual student is not able to justify his/her project, then the project will be kept under plagiarism.

NOTE: The technicality of the project will be judged during the demo and marked accordingly.

Marking Breakdown for Coursework	
Criteria	**Total Marks**
1. Introduction	*5*
2. Algorithm	*5*
3. Pseudocode	*5*
4. Flowchart	*5*
5. Data Structures	*5*
6. Program	
a. Correctness	*10*
b. Implementation (modularity, use of functions/classes)	*10*
c. Programming Style	*10*
d. Exception Handling	*10*
e. User interface/program usability	*10*
7. Testing	*5*
8. Conclusion	*5*
9. VIVA	*10*
10. Report Structure and Formatting	*5*
Total:	**100**