LONDON METROPOLITAN UNIVERSITY

1st SIT COURSEWORK Question Paper: **Summer Semester 2021**

Module Code:	**CS4051NI**
Module Title:	**Fundamentals of Computing**
Module Leader:	**Sukrit Shakya** (Islington College)

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **60%** of your total module grades.
Submission Date:	**Week 12**
When Coursework is given out:	**Week 6 (28th July, 2021)**
Submission Instructions:	Submit the following to Islington College RTE department 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.

© London Metropolitan University

Plagiarism Notice

You are reminded that there exist regulations concerning plagiarism.

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 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:

(i) 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.
(ii) 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 = mc2 (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.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from http://www.londonmet.ac.uk/academic-regulations

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 ***library management system***
- ✓ Describe the program
- ✓ Test the program with some sample data to demonstrate its behavior
- ✓ Write a report to present the work

- Students are required to submit weekly assignment along with related 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).

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

1. Scenario

A library maintains its book information (stock) in a text file. An application needs to be developed which will read the text file and display all the books available for borrowing. Then with each transaction (borrowing) a note should be generated for the particular borrower and should be written into a file. The stock of the books should also be updated after each transaction. For example, if the library had 10 books of specific title, then if one of those books is borrowed then the number should change to 9. In the case of returning a book, a note should again be generated for the person returning the book. The stock should also be updated i.e. the quantity of the book returned should be increased by 1.

A sample format of the text file containing the information about the books is as follows:

```
Harry Potter,JK Rowling,30,$2
Start With Why,Simon Sinek,10,$1.5
Programming With Python,John Smith,20,$1.5
```

**1st column contains the name of the book, 2nd column contains the name of the author, 3rd column contains the quantity available, 4th column contains the price for borrowing the book (for duration of 10 days)*

***You can use your own format and add other information too*

A note should be generated for each transaction. When a person borrows a book a note should be generated which must contain the name of the borrower, the name of the book borrowed, the date and time of such issue and the total amount to be paid for the borrowing. If a person decides to borrow multiple books then all the borrowed books should be written to the note and the amount should be added up for all the borrowed books.

When a person returns a book to the library, a note should be generated and written to the file again. The note should contain the name of the borrower, name of the book, date and time of the return. The lending duration should be set to 10 days, and if a person is late in returning the book, a fine should be applied on a daily basis which should also be present in the generated file.

** The format of the borrowing and returning notes is up to you. But each file should have a unique name.*

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, data storage.
    - It can use any primitive or complex data structures which might be necessary for holding the data (lists, tuples, 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 books and waiting for the administrator to enter the details. 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 user input, reading files, generating borrowing/returning 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 screen shots) that has been carried out for the program. There should be at least 5 various test cases following a specific testing technique.

        - The report must include a **Conclusion** section where they need to present their finding of the development and research (approx. 300 words).

Marking Breakdown for Coursework	
Criteria	**Total Marks**
1. Introduction	*10*
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. Programing Style	*10*
d. Exception Handling	*10*
e. User interface/program usability	*10*
7. Testing	*10*
8. Conclusion	*5*
9. Report Structure and Formatting	*5*
Total:	**100**