



## Assignment

<b>Faculty Name:</b>	Information Technology
<b>Module Code:</b>	ITDPA2-B44
<b>Module Name:</b>	Data Structures and Algorithms in Python Block 4
<b>Module Leader:</b>	Mr Dennis Owuor
<b>Internal Moderator:</b>	Dr Amos Anele Dr Michael Ajayi
<b>Copy Editor:</b>	Mr Kevin Levy
<b>Total Marks:</b>	80
<b>Submission Date:</b>	22/11/2021 – 26/11/2021

---

This module is presented on NQF level 6.

5% will be deducted from the student's assignment mark for each calendar day the assignment is submitted late, up to a maximum of three calendar days. The penalty will be based on the official campus submission date.

Assignments submitted later than three calendar days after the deadline or not submitted will get 0%.<sup>1</sup>

This is an individual assignment.

---

**This assignment contributes 40% towards the final mark.**

### Instructions to Student

1. Remember to keep a copy of all submitted assignments.
2. All work must be typed.
3. Please note that you will be evaluated on your writing skills in all your assignments.
4. All work must be submitted through Turnitin<sup>2</sup> and the full Originality Report should be attached to the final assignment. Negative marking will be applied if you are found guilty of plagiarism, poor writing skills or if you have applied incorrect or insufficient referencing.

---

<sup>1</sup> Under no circumstances will assignments be accepted for marking after the assignments of other students have been marked and returned to the students.

<sup>2</sup> Refer to the PIHE Policy for Intellectual Property, Copyright and Plagiarism Infringement, which is available on myLMS.

(See the table at the end of this document where the application of negative marking is explained.)

5. Each assignment must include a cover page, table of contents and full bibliography, based on the referencing method applicable to your faculty as applied at Eduvos.
6. Use the cover sheet template for the assignment; this is available from *myLMS*.
7. Students are not allowed to offer their work for sale or to purchase the work of other students. This includes the use of professional assignment writers and websites, such as Essay Box. If this should happen, Eduvos reserves the right not to accept future submissions from a student.
8. Students must submit their code alongside the answer sheet.

### **Assignment Format**

Students must follow the requirements when writing and submitting assignments as follows:

- Use Arial, font size 10.
- Include page numbers.
- Include a title page.
- Print submissions on both sides of the page.
- Write no more than the maximum word limit.
- Ensure any diagrams, screenshots and PowerPoint presentations fit correctly on the page and are referenced.
- Include a table of contents.
- Use the accurate referencing method throughout the assignment.
- Include a bibliography based on the applicable referencing method at the end of the assignment.
- Include the completed Assessment/Project Coversheet (available on *myLMS*).
- Check spelling, grammar and punctuation.
- Run the assignment through Turnitin software.

### **Essential Embedded Knowledge and Skills Required of Students**

- Report-writing skills
- Ability to analyse scenarios/case studies
- Understanding of subject field concepts and definitions
- Ability to apply theoretical knowledge to propose solutions to real-world problems
- Referencing skills

## Resource Requirements

- A device with Internet access for research
- A desktop or personal computer for typing assignments
- Access to a library or resource centre
- Prescribed reading resources

## Delivery Requirements (evidence to be presented by students)

- A typed assignment<sup>3</sup>
- A Turnitin Originality Report

## Minimum Reference Requirements

At least five references for first year, ten references for second year and fifteen references for third year.

Additional reading is required to complete this assignment successfully. You need to include the following additional information sources:

- Printed textbooks/e-books
- Printed/online journal articles
- Academic journals in electronic format accessed via PROQUEST or other databases
- Periodical articles e.g. business magazine articles
- Information or articles from relevant websites
- Other information sources e.g. geographic information (maps), census reports, interviews

### Note

- It is crucial that students reference all consulted information sources, by means of in-text referencing and a bibliography, according to the applicable referencing method.
- Negative marking will be applied if a student commits plagiarism i.e. using information from information sources without acknowledgement and reference to the original source.
- In such cases, negative marking, also known as 'penalty scoring', refers to the practice of subtracting marks for insufficient/incorrect referencing.
- Consult the table at the end of this document, which outlines how negative marking will be applied as well as the way in which it will affect the assignment mark.

---

<sup>3</sup> Refer to the Conditions of Enrolment for more guidance (available on myLMS).

---

## Section A

---

### Learning Objective

The assignment aims to ensure students have gained the required technical knowledge and practical skills in data structures and algorithms with Python and can apply it to solve related real-world problems.

### Assignment Topic

The assignment requires students to demonstrate an understanding of:

- Data structure fundamentals in Python
- Stacks, queues, dequeues, linked lists and recursion
- Trees and priority queues
- Maps, hashing and search trees
- Sorting and selection

### Scope

The scope of this assignment is focused on lecture weeks 1 to 5

### Technical Aspects

Scientific computing using the Python programming language (Computer with Anaconda IDE)

Study the scenario and complete the question(s) that follow:

### Patient's Record

Steve Biko Hospital is in the process of storing her patient's data in a format stipulated in Table 1.1 below which will, in turn, be uploaded to a database. Note that the patient numbers are arranged in ascending order.

The hospital office administrator uses a specific format to capture the patient's data in a text file in no particular order. Suppose the specific format structure is defined line by line where each patient's data consists of the following: patient number, surname, name, age group, sex, and weight, which are captured in this order.

The age groups are given with descriptions as follows: Neonate (first 4 weeks of life), Infant (1 month to 1 year), Toddler (1 – 3 years), Pre-school child (3 – 5 years), School Age Child (6 – 12 years), Adolescent (13 – 18 years), Young Adult (19 – 40), Middle Age Adult (41 – 65 years), and Older Adult (over 65 years) which are often represented by digits 1 to 9 (where 1 is Neonate and 9 is Older Adult).

You are expected to develop a Python program that will produce the output as shown Table 1.1 by creating a text file using the format described above, which will serve as the input to your program.

PATIENT DATA					
PATIENT NO.	SURNAME	NAME	AGE GROUP	SEX	WEIGHT(KG)
10005	Linda	Khumalo	Young Adult	F	56.40
10012	Mokoena	Mkhize	School Age Child	M	9.82
10015	Dlamini	Nkosi	Infant	M	5.60
10100	Sithole	Mahlangu	Older Adult	M	91.77
10132	Ntombi	Zulu	Pre-school child	F	10.61
Number of patients: 5					

**Table 1.1: Patient Data**

Source: Owuor, D.L. (2021)

Using Object-Oriented Programming Abstract Data Types (ADT) principles in Python:

1.1 Create a **class "PatientFileReader ()"** for retrieving the patient data from the given text file (this should have been created by the student). (2 Marks)

1.2 Create a **function "open ()"** to open a connection to the text file. (1 Mark)

- 1.3 Create a **function “close ()”** to close a connection to the text file. (1 Mark)
- 1.4 Create a **function “getAll ()”** that will retrieve all the patient data and store them in a list. (3 Marks)
- 1.5 Create a **function “getData ()”** that will retrieve the next patient data from the text file. The function must return “None” when there is no additional data to be retrieved. (5 Marks)
- 1.6 Create a **class “PatientData ()”** to store all the individual patient data. (3 Marks)
- 1.7 Create a **function “PrintData ()”** to print the exact information depicted in Table 1.1. Note that the patient numbers are arranged in ascending order. (2 Marks)
- 1.8 Create a **function “main ()”** to execute the entire program. It must also sort the list created in “d” by the patient number. (3 Marks)

**PS:** Students must comment on the code to prove that they understand the implementation. Also, the text file created, and the source code should be submitted.

**[Sub Total 20 Marks]**

---

End of Question 1

Study the scenario and complete the question(s) that follow:

**Digital Electronic Store**

A recently graduated programmer, Nkosi, has been contracted to build a system for a digital electronic store situated in Pretoria, Gauteng, South Africa. Considering the number of transactions needed, the ability to sort and search for data may be an integral feature that must be added to the system. Even so, the pattern of the stored data may determine the programming concept employed.

Considering your knowledge in this area, you will be required to demonstrate an implementation of sorting and searching algorithms given an array of data. Your concept is expected to assist the programmer in developing the application.

Source: Owuor, D.L. (2021)

Using the merge sort and Linked List ADT principles:

- 2.1 Write a Python program that will accept inputs for the purpose of outputting two arrays. The first input must accept 6 integer numbers which will output an array of all even numbers from 12 to 1 arranged in an unsorted manner while the second input must accept 4 integer numbers which will output an array of all odd numbers between 20 and 40 arranged in an unsorted manner. (3 Marks)
- 2.2 Write a Python program using merge sort to sort the unsorted arrays in Question 2.1. Your program must first accept the two unsorted arrays, merge them, and then return a sorted array. (15 Marks)
- 2.3 A linked list is like an array. However, they differ in some sense based on their chained-like element storing address location of next elements. This presents an advantage over arrays. Equipped with this knowledge, you are bent on providing some advice to Nkosi by implementing a singly linked list using the merged sorted array in Question 2.2. Hence, write a Python program that will take the merged sorted array in Question 2.2 as input, convert it to a singly linked list and:
- a. Print it in a singly linked list format in the same order as the input (5 Marks)
  - b. Print it in a singly linked list format in a reverse order of the input. (5 Marks)

- c. Insert the value “45” after the last value in the linked list (reverse order) using the value as a reference and not the index. Print the entire linked list as a proof to show it was appropriately inserted. (2 Marks)

The output should be as below:

```
Linked list in same order as merged sorted arrays:
2-->4-->6-->8-->10-->12-->23-->29-->31-->37-->
Linked list in reverse order as merged sorted arrays:
37-->31-->29-->23-->12-->10-->8-->6-->4-->2-->
Linked list in reverse order of merged sorted arrays after the value 45 is added:
37-->31-->29-->23-->12-->10-->8-->6-->4-->2-->45-->
```

**[Sub Total 30 Marks]**

---

End of Question 2



Study the scenario and complete the question(s) that follow:

### Grocery Store

Suppose you have been asked to build an e-commerce website for a grocery store with a given number of specific items required to be sold by the store. The CEO of the store gave you a brief of the available items and how they will be arranged in the store located in one of their branches named “Building 1”. An extract of the brief given shows that the groceries were categorised as follows: pastries, fruits, vegetables, and meat. Each category is highlighted in Figure 1.1. If “Building 1” is divided into four (4) sections namely, South, East, West, and North with each category of groceries located in these sections, i.e. pastries (South), fruits (East), vegetables (West), and ,eat (North). Each item can also be further identified by the shelf number as indicated in Figure 1.3.

```
Groceries
|__Pastries
|   |__Bread
|   |__Cake
|   |__Cheese Griller
|   |__Meat Pie
|__Fruits
|   |__Apples
|   |__Oranges
|   |__Mangoes
|   |__Pawpaw
|__Vegetables
|   |__Broccoli
|   |__Carrots
|   |__Potatoes
|   |__Okra
|__Meat
|   |__Beef
|   |__Chicken
|   |__Turkey
|   |__Pork
```

Figure 1.1: Groceries and Specific Items

```

Building 1
|__South
|   |__South - Shelf 1
|   |__South - Shelf 2
|   |__South - Shelf 3
|   |__South - Shelf 4
|__East
|   |__East - Shelf 1
|   |__East - Shelf 2
|   |__East - Shelf 3
|   |__East - Shelf 4
|__West
|   |__West - Shelf 1
|   |__West - Shelf 2
|   |__West - Shelf 3
|   |__West - Shelf 4
|__North
|   |__North - Shelf 1
|   |__North - Shelf 2
|   |__North - Shelf 3
|   |__North - Shelf 4

```

**Figure 1.2: Buildings and Locations**

```

Groceries (Building 1)
|__Pastries (South)
|   |__Bread (South - Shelf 1)
|   |__Cake (South - Shelf 2)
|   |__Cheese Griller (South - Shelf 3)
|   |__Meat Pie (South - Shelf 4)
|__Fruits (East)
|   |__Apples (East - Shelf 1)
|   |__Oranges (East - Shelf 2)
|   |__Mangoes (East - Shelf 3)
|   |__Pawpaw (East - Shelf 4)
|__Vegetables (West)
|   |__Broccoli (West - Shelf 1)
|   |__Carrots (West - Shelf 2)
|   |__Potatoes (West - Shelf 3)
|   |__Okra (West - Shelf 4)
|__Meat (North)
|   |__Beef (North - Shelf 1)
|   |__Chicken (North - Shelf 2)
|   |__Turkey (North - Shelf 3)
|   |__Pork (North - Shelf 4)

```

**Figure 1.3: Grocery and Item Locations**

Items	Prices
Bread	14.10
Cake	22.20
Meat Pie	12.90
Cheese Griller	23.60
Apples	2.10
Oranges	1.20
Mangoes	2.60
PawPaw	2.90
Broccoli	10.20
Carrots	4.30
Potatoes	99.90
Okra	60.50
Beef	100.20
Chicken	210.10
Turkey	102.11
Pork	80.35

**Table 1.2: Price Value per Item**

Source: Owuor, D.L. (2021)

Based on the scenario:

- 3.1 Write a Python program that will represent the grocery items and their locations in a general tree format using the Tree ADT Principles. The program must also print the exact output of Figures 1.1, 1.2 and 1.3. (15 Marks)
  
- 3.2 Suppose while creating the groceries website, you want each category in the grocery store to be represented by a page. Implement a Python program that will allow you demonstrate the LIFO action using stack collections.deque. (5 Marks)

3.3 A Hash Map or Hash Table allows values to be stored in key value pairs and hence by using the key, the value is easily retrieved. Suppose the items in the grocery store have a particular price value which are meant to be assigned to them respectively as in Table 1.2. Using a Hash Table and simple hash function (division method), implement a Python program that will assign the prices to the various items as well as the **memory location** it was stored. In addition, the application should be able to retrieve the price values, delete and set values. (10 Marks)

**[Sub Total 30 Marks]**

---

End of Question 3

---

## Section B

---

### **Plagiarism and Referencing**

Eduvos places high importance on honesty in academic work submitted by students and adopts a zero-tolerance policy on cheating and plagiarism. In academic writing, any source material, e.g. journal articles, books, magazines, newspapers, reference material (dictionaries), or online resources (websites, electronic journals, or online newspaper articles), must be properly acknowledged. Failure to acknowledge such material is considered plagiarism; this is deemed an attempt to mislead and deceive the reader and is unacceptable.

Eduvos adopts a zero-tolerance policy on plagiarism; therefore, any submitted assessment that has been plagiarised will be subject to severe penalties. Students who are found guilty of plagiarism may be subject to disciplinary procedures, and outcomes may include suspension from the institution or even expulsion. Therefore, students are strongly encouraged to familiarise themselves with referencing techniques for academic work. Students can access the Eduvos referencing guides on *myLMS*.

---

## Negative Marking

---

### Second-year Students

- A minimum of 10 additional information sources must be consulted and correctly cited.
- If no additional information sources have been used, a full 10% must be deducted.
- Deduct 1% per missing resource of the required 10. For example:
  - If only five resources cited, deduct 5%.
  - If only three resources cited, deduct 7%.
- Markers to apply the penalties for Category A for insufficient sources and incorrect referencing style.
- To determine the actual overall similarity percentage and plagiarism, markers must interpret the Turnitin Originality Report with reference to credible sources used and then apply the penalties as per the scale in the Policy for Intellectual Property, Copyright and Plagiarism Infringement.
- The similarity report alone is not an assessment of whether work has or has not been plagiarised. Careful examination of both the submitted paper/assignment/project and the suspect sources must be done.

### Category A

Minimum Reference Requirements	Deduction from Final Mark
No additional information sources have been used or referenced.	10%