
Principles of Programming

CT4029

Workbook – Session 6

2022-23

Zainab Loukil

Qublai Khan Ali Mirza

School of Computing and Engineering

University of Gloucestershire

2022/23

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, including – but not limited to – photocopy, recording, or any information storage and retrieval system, without the specific prior written permission of University of Gloucestershire.

In this session we covered data structures in python. We specifically focused on how different types of data structures help programmers to design their algorithms to efficiently solve a problem. This section is comprised of different questions, which require appropriate data structures to be used to solve problems.

Following are the Learning Outcomes of this Section:

- **Managing Data Efficiently**
- **List Operations**
- **Tuples Operations**



1. Section A

1.1. Exercise 1

Write a program that takes two words as input from user and match them to check whether the words are an anagram or not.

Note: An anagram is a word formed by rearranging the letters of a different word, typically using all the original letters exactly once. Example: Elbow = Below, State = Taste, Cider = Cried

Hint: Strategy to identify anagrams is to take the letters of a word, sort those letters, then compare the sorted sequences. Anagrams should have the same sorted sequence.

1.2. Exercise 2

Write a program to count all the elements present in a list until the elements is a tuple.

1.3. Exercise 3

Write a Python program to get the frequency of the elements in a list

1.4. Exercise 4

Write a Python program to find common items from two lists.

1.5. Exercise 5

Write a Python program to remove duplicates from a list of lists.

Sample list: `[[50, 30], [60], [30, 56, 25], [50, 30], [23], [40]]`

New List: `[[50, 30], [60], [30, 56, 25], [23], [40]]`

1.6. Exercise 6

Write a menu-based (console-based) program to manage students' marks and grades. your program should ask user to select a module-> assessment -> and enter marks for the assessment. Once you have entered the marks, program should be able to present grades for each module, based on the grading system. User should be able to enter the

CAT points for each module (15 – 30 CATS), if it's a 30 CAT module then there should be two assessments and the average two should be the final marks of that module. The program should also be able to present a final transcript of marks and grades with a final percentage.