

# DSW03A1 - Development Software

06<sup>th</sup> March 2023  
Course Lecturer: Sandile Thamie Mhlanga



## Lab Exercise 2: Big Data and File Manipulation

### Instructions

The concepts included have been covered in class under LU 1 Big Data, IoT and 4IR Technologies.

**This Lab Ex is a Face-to-Face session. Submission is required. Attendance and Submission counts for marks. TUT submitted remotely without attendance will attract zero marks.**

- **IMPORTANT:** Please name your Visual Studio Solution like so (failure to do so will attract a **1%** penalty):
  - [STUDENT\_NO]\_[SURNAME]\_LabEx2
  - (e.g. 222833200\_Mhlanga\_LabEx2)
- **PLAGIARISM:** Please refer to your Learning Guide as well as the latest University of Johannesburg's plagiarism policy document entitled: "POLICY: PLAGIARISM"

**COPYING:** This is an individual assignment; if any copying is detected, all parties involved will score a **0%** for the assignment and **WILL** face disciplinary consequences

### Question 1

#### Background

Recall from LU1 that Big Data is often **large volume of data** – **both structured and unstructured**; and that it **may be analysed computationally to reveal patterns, trends, and associations**; keep this mind as you engage through this exercise.

Search the internet for a dataset of Big Data (you need to be able to explain why you think the dataset you have selected can be classified as Big Data).

Once you find this data set:

- Store this information in a text file
- Create a C# program to read this file in and store the information in a data structure.
- The user must be able to display the data based on two categories e.g. either year and level in the example given below.

The application must provide the user with menu option as shown in **Figure 1**

```
*****
*                                *
*                                *
*****
1: Display all data from the dataset
2: Display all data based on the given year (2013 - 2018)
3: Display all data based on the level (1 - 4)
0: Exit.....
```

### Figure 1

**Option 1** must display all the data stored in the file.

**Option 2** must display all the data based on the supplied year.

**Option 3** must display the data based on another or different criteria. In the given example, it is based on the “Level”

**Option 0** must terminate the program

```

2013,Level 3,ZZ11,Food product manufacturing,Dollars (millions),H21,Opening stocks,Financial performance,3722,"ANZSIC06 groups C111, C112, C113, C114, C115, C116, C117, C118, C119, C120, C121, C122, C123, C124, C125, C126, C127, C128, C129, C130, C131, C132, C133, C134, C135, C136, C137, C138, C139, C140, C141, C142, C143, C144, C145, C146, C147, C148, C149, C150, C151, C152, C153, C154, C155, C156, C157, C158, C159, C160, C161, C162, C163, C164, C165, C166, C167, C168, C169, C170, C171, C172, C173, C174, C175, C176, C177, C178, C179, C180, C181, C182, C183, C184, C185, C186, C187, C188, C189, C190, C191, C192, C193, C194, C195, C196, C197, C198, C199, C200, C201, C202, C203, C204, C205, C206, C207, C208, C209, C210, C211, C212, C213, C214, C215, C216, C217, C218, C219, C220, C221, C222, C223, C224, C225, C226, C227, C228, C229, C230, C231, C232, C233, C234, C235, C236, C237, C238, C239, C240, C241, C242, C243, C244, C245, C246, C247, C248, C249, C250, C251, C252, C253, C254, C255, C256, C257, C258, C259, C260, C261, C262, C263, C264, C265, C266, C267, C268, C269, C270, C271, C272, C273, C274, C275, C276, C277, C278, C279, C280, C281, C282, C283, C284, C285, C286, C287, C288, C289, C290, C291, C292, C293, C294, C295, C296, C297, C298, C299, C300, C301, C302, C303, C304, C305, C306, C307, C308, C309, C310, C311, C312, C313, C314, C315, C316, C317, C318, C319, C320, C321, C322, C323, C324, C325, C326, C327, C328, C329, C330, C331, C332, C333, C334, C335, C336, C337, C338, C339, C340, C341, C342, C343, C344, C345, C346, C347, C348, C349, C350, C351, C352, C353, C354, C355, C356, C357, C358, C359, C360, C361, C362, C363, C364, C365, C366, C367, C368, C369, C370, C371, C372, C373, C374, C375, C376, C377, C378, C379, C380, C381, C382, C383, C384, C385, C386, C387, C388, C389, C390, C391, C392, C393, C394, C395, C396, C397, C398, C399, C400, C401, C402, C403, C404, C405, C406, C407, C408, C409, C410, C411, C412, C413, C414, C415, C416, C417, C418, C419, C420, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441, C442, C443, C444, C445, C446, C447, C448, C449, C450, C451, C452, C453, C454, C455, C456, C457, C458, C459, C460, C461, C462, C463, C464, C465, C466, C467, C468, C469, C470, C471, C472, C473, C474, C475, C476, C477, C478, C479, C480, C481, C482, C483, C484, C485, C486, C487, C488, C489, C490, C491, C492, C493, C494, C495, C496, C497, C498, C499, C500, C501, C502, C503, C504, C505, C506, C507, C508, C509, C510, C511, C512, C513, C514, C515, C516, C517, C518, C519, C520, C521, C522, C523, C524, C525, C526, C527, C528, C529, C530, C531, C532, C533, C534, C535, C536, C537, C538, C539, C540, C541, C542, C543, C544, C545, C546, C547, C548, C549, C550, C551, C552, C553, C554, C555, C556, C557, C558, C559, C560, C561, C562, C563, C564, C565, C566, C567, C568, C569, C570, C571, C572, C573, C574, C575, C576, C577, C578, C579, C580, C581, C582, C583, C584, C585, C586, C587, C588, C589, C590, C591, C592, C593, C594, C595, C596, C597, C598, C599, C600, C601, C602, C603, C604, C605, C606, C607, C608, C609, C610, C611, C612, C613, C614, C615, C616, C617, C618, C619, C620, C621, C622, C623, C624, C625, C626, C627, C628, C629, C630, C631, C632, C633, C634, C635, C636, C637, C638, C639, C640, C641, C642, C643, C644, C645, C646, C647, C648, C649, C650, C651, C652, C653, C654, C655, C656, C657, C658, C659, C660, C661, C662, C663, C664, C665, C666, C667, C668, C669, C670, C671, C672, C673, C674, C675, C676, C677, C678, C679, C680, C681, C682, C683, C684, C685, C686, C687, C688, C689, C690, C691, C692, C693, C694, C695, C696, C697, C698, C699, C700, C701, C702, C703, C704, C705, C706, C707, C708, C709, C710, C711, C712, C713, C714, C715, C716, C717, C718, C719, C720, C721, C722, C723, C724, C725, C726, C727, C728, C729, C730, C731, C732, C733, C734, C735, C736, C737, C738, C739, C740, C741, C742, C743, C744, C745, C746, C747, C748, C749, C750, C751, C752, C753, C754, C755, C756, C757, C758, C759, C760, C761, C762, C763, C764, C765, C766, C767, C768, C769, C770, C771, C772, C773, C774, C775, C776, C777, C778, C779, C780, C781, C782, C783, C784, C785, C786, C787, C788, C789, C790, C791, C792, C793, C794, C795, C796, C797, C798, C799, C800, C801, C802, C803, C804, C805, C806, C807, C808, C809, C810, C811, C812, C813, C814, C815, C816, C817, C818, C819, C820, C821, C822, C823, C824, C825, C826, C827, C828, C829, C830, C831, C832, C833, C834, C835, C836, C837, C838, C839, C840, C841, C842, C843, C844, C845, C846, C847, C848, C849, C850, C851, C852, C853, C854, C855, C856, C857, C858, C859, C860, C861, C862, C863, C864, C865, C866, C867, C868, C869, C870, C871, C872, C873, C874, C875, C876, C877, C878, C879, C880, C881, C882, C883, C884, C885, C886, C887, C888, C889, C890, C891, C892, C893, C894, C895, C896, C897, C898, C899, C900, C901, C902, C903, C904, C905, C906, C907, C908, C909, C910, C911, C912, C913, C914, C915, C916, C917, C918, C919, C92
```

Criteria	Weight (%)
Ability to read in the input file and display all data	5
Ability to read in the input file and display data based on the year	6
Display data based on another criteria beside the year	8
Menu created and works correctly	3
User interface: Main	10
Neatness, good programming principles (including variable naming, comments, etc.)	3
<b>TOTAL</b>	<b>35</b>

## Question 2

### Background (Windows-based application and Arrays of Objects)

You are requested to develop a C# application for our institution. The application determines a student's final mark based on the marks obtained from the assignment, test and examination. The assignment, test, and exam have the weights of 20%, 30% and 50% respectively. The captured user input and the computed final mark are stored in a data structure such as an array of objects. The application must also save all students' data in a "TXT" file called **exam.txt**. The below diagrams illustrate the operations that must be performed

UJ Portal

University Of Johannesburg Exam Portal

Student Number: 2022123

Assignment: 65

Test: 45

Exam: 52


Final: 52.5

Student Number	Final Mark	ResultsCode
----------------	------------	-------------

Buttons: Compute, Add, Display, Write, Clear, Exit

Figure 1

UJ Portal

 **University Of Johannesburg Exam Portal**

Student Number

Assignment

Test

Exam

Final

Compute Add Display Write Clear Exit


Student Number Final Mark ResultsCode

Record 1 has been added to the list

OK

Figure 2

UJ Portal

 **University Of Johannesburg Exam Portal**

Student Number

Assignment

Test

Exam

Final

Compute Add Display Write Clear Exit

Student Number	Final Mark	ResultsCode
2022123	52,5	Pass
2022785	34	Fail
2022556	73	Pass

Figure 3

## Exercise

Based on the given background do the following:

- Create a window form interface with the various GUI controls has shown in **Figure 1**
- Create class' that are need to solve the given problem, you can have more the one class, for example:
  - **Student class**: contains the student attributes such as student number, assignment mark, test mark, etc.
  - **StudentRecord class**: stores student's objects in a data structure.
- The application must also store the data in the data structure into a file.

## Rubric

	Criteria	Max	Mark
	GUI Interface	9	
	Ability to create class student	5	
	Ability to create StudentRecord Class with a data structure e.g array of objects	5	
	Ability for the main to create and invoke various methods	5	
	Read to file	6	
	TOTAL	30	

**NB: This Lab Ex must be submitted by 23:59, Monday 13th of March 2023**