UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS110 - Program Design: Introduction
Practical 1

1 Introduction

Deadline: 27th August, 19:30

1.1 Objectives and Outcomes

The objective of this practical is to test your understanding of the programming concepts covered in the theory classes and as revision for COS132. In particular, this practical will test your understanding of pointers and 2D dynamic arrays.

1.2 Submission

All submissions are to be made to the **ff.cs.up.ac.za** page under the COS 110 page, and for the correct practical slot. Submit your code to Fitchfork before the closing time. Students are **strongly advised** to submit well before the deadline as **no late submissions will be accepted**.

1.3 Plagiarism

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent) and copying textual material from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to **http://www.ais.up.ac.za/plagiarism/index.htm** (from the main page of the University of Pretoria site, follow the *Library* quick link, and then click the *Plagiarism* link). If you have questions regarding this, please ask one of the lecturers, to avoid any misunderstanding.

1.4 Implementation Guidelines

Follow the specifications of the practical precisely. For each practical, you will be required to create your own makefile so pay attention to the names of the files you will be asked to create. If the practical requires you to submit additional files of your own, follow the file structure and format exactly. Incorrect submissions will use up your uploads and no extensions will be given. In terms of C++, unless otherwise stated, the usage of C++11 or

additional libraries outside of those indicated in the practical, will not be allowed. Some of the appropriate files that you submit will be overwritten during marking to ensure compliance to these requirements.

1.5 Mark Distribution

Activity	**Mark**
Task 1	20
Total	**20**

2 Practical

2.1 Task 1

For this task you are asked to write a program that consists of a file **recap.cpp** and a **makefile** to compile and run it. You will need to include a blank **data.txt** file in your submission as well. Your makefile must compile the executable as **main**.

You will write all of your code inside **recap.cpp** including the main and any required headers etc.

At the start of your program, your code will open and read values in from a file **data.txt**. The file will have an unknown number of lines. You are required to read this data, line by line, into a 2D dynamic array of strings.

The format of the line is as follows:

```
1,a,b,c,d
```

Each line starts with an ID, which is an int, and then will be followed by an unknown number of strings separated by commas. Each string in the line is associated with the ID of the line.

Your 2D dynamic array should be sized to exactly the size of the number of items in each line. Each line will have at least the ID and 1 item.

Once the items are stored in the array you need to print out the array in ascending order based on the order IDs in the following format:

```
ID,a,b,c
```

An example of this output is as follows (for 2 orders):

```
3,toaster,iron
4,spanner,wrench,hammer
```

Pay attention to the spaces and formatting requirements. Terminate each line of output with a newline. Remember to deallocate any memory you have used at the end of your program.

The following libraries are allowed:

- iostream
- fstream
- string
- sstream

You will have a maximum of 10 uploads for this task.