Department of Computer Science
Practical 1 2016

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

1 Introduction

1.1 Deadline

Complete all the tasks in this assignment before **Friday the** 19^{th} **of February 2014 at 19:30**.

1.2 Objectives and Outcomes

This practical assignment introduce students to the basics of on-campus email, basic Linux utilities including the creation and extraction of tar archives, compiling a simple program using the g++ compiler, uploading to the Computer Science website and using Fitchfork. Students will also be required to use the RT ticket system.

1.3 Mark allocation

Activity	**Mark**
Book a lab session	2
Email to the RT system	5
Create a directory structure	5
Upload a C++ Program	8
Total	**20**

1.4 Login Details

There are two sets of login details that you will use for this module and probably also other modules to follow. The first set is your **University of Pretoria (UP) login details**, and the second is your **Computer Science (CS) login details**. You should know your login details. More information about these and where to get them can be found in the notes called `Login101.pdf` on the ClickUP site.

2 The CS Website

You can access the site by going to `http://www.cs.up.ac.za/courses/COS132`.

1. If you read this you probably downloaded this document from the CS website. If you retrieved it by any other means, please go to the CS website and download it.
2. Login to the CS website to ensure your username and password is working.

3. Familiarize yourself with the lecturers by visiting their profiles.

4. For the practicals to come, you are required to book a lab session. Students will **not be allowed** to attend a practical session if they didn't book. Since lab space is limited, the booking works on a first-come-first-serve basis.

3 The UP Email Facility

As a university of Pretoria you are automatically assigned an Gmail account. If you are not familiar with your UP email account, your first task is to read `UP_mail.pdf`.

1. Play around with your email account and check which settings you can tune.

2. Send an email to cos132queries@cs.up.ac.za. The subject line should be **exactly** as follows:

   ```
   Introducing uXXXXXXXX to COS 132
   ```

   where XXXXXXXX is your 8-digit student number. Remember the "u" in front of your student number. Include your name, email address and student number in the message body (format doesn't matter). Make sure you send this email from your Tuks email account. You will receive an email response. Allow a couple of minutes for the system to respond.

4 Files and directories

Refer to Chapter 5 of *Tricks of the Trade for Novice Programmers Volume 1* for notes on how to perform the tasks in this section.

1. Download the archive `Prac1.tar.gz` from the CS website.

2. Extract the archive.

3. Create directories and move the files you have downloaded to produce the following directory structure in your home directory:

   ```
   COS132
   └── Prac1
       ├── GivenDocs
       │   ├── Practical1.pdf
       │   └── Prac1.tar.gz
       └── HelloProgram
           ├── hello.cpp
           └── makefile
   ```

5 Text Editors and Compiling

5.1 Text Editors

In order to write a program in C++ you will need a text editor. There are a number of different editors available, such as SciTE, gedit, Kate and KDevelop. It is up to you which one to use, but we recommend SciTE, since it is cross-platform (Windows, Linux, Mac OSX). To open SciTE click on the  icon. Alternatively you can use the following command:

```
scite <file to open> &
```

Linux is case sensitive. Even though the program is called SciTE, you should use `scite` as the command. If you don't specify a file to open, SciTE will open and show a blank document. To open a file called `file.txt`, use the following command:

```
scite file.txt &
```

5.2 Compiling

In order for a computer to execute a program, the human-readable code has to be converted into a binary format that computers can understand. This process is called "compiling". Note that this binary format differs between operating systems, hence C++ code compiled in Windows will not work in Linux and vice versa. We will be using the GNU GCC compiler for compilation in Linux. The compiler can be invoked by the `g++` command. You will learn how exactly to use this command in the practicals to come.

1. Open the file `hello.cpp` we provided in SciTE and look at what the C++ code snippet does. Change the program to output the following:

   ```
   My student number is XXXXXXXX and I think programming ...
   ```

   `XXXXXXXX` represents your student number without the leading `u`. You may replace ... with any word or phrase of your choice, but your output should end with a fullstup or exlamation mark.

2. Make sure that you saved the changes you made to `hello.cpp`.

3. Compile the program by using the following command. You have to open a command terminal and be in the directory where `hello.cpp` is located when typing the command.

   ```
   g++ hello.cpp -o hello.out
   ```

   You can then run the program you just compiled with the following command:

   ```
   ./hello.out
   ```

6 Test

To test the program you should compile and run the program using the following command:

```
make run
```

Remove any errors before uploading to fitchfork.

7 Upload and Fitchfork

Fitchfork is a automated marking system developed inhouse at the Computer Science Department of the University of Pretoria. Once you upload a practical assignment, Fitchfork will automatically compile your program and run a couple of tests. It will then give you a mark based on how many test cases were successful. Since Fitchfork is not a human being it will only follow the strict instructions that we provide. Hence, if the practical assignment states that you have to create a file `practical1.cpp` and you named it `Practical1.cpp`[1], Fitchfork won't be able to compile it and you won't get any marks. Always make sure you strictly follow the instructions we provide.

If Fitchfork wasn't able to compile your program or detected some problems in your code it will provide you with some hints and guidance. Once uploaded you can view your Fitchfork marks and the corresponding feedback by going to your `My Automarks` on the CS website.

You are advised to test and debug your code thoroughly before uploading to Fitchfork. If Fitchfork was unable to assess your task (give you -1), you should use the Fitchfork feedback to correct the error before uploading again.

The number of resubmissions for the same task is limited. This number is shown on the dialog where you upload your tarball.

1. create an TAR archive containing only the `hello.cpp` you edited. Give the archive a descriptive name, such as `COS132Prac1.tar`.
2. Upload your TAR archive to the Practical 1 slot on the CS website.
3. Go to `My Automarks` and view the marks for your upload. This Fitchfork assignment is out of 8.
4. You will be allowed 10 tries.

8 Finishing Off

To avoid abuse of your account, you are advised to log off before leaving the lab. Since other people using the computer afterwards will probably not use Linux, it is always a good idea to restart the computer. It will then automatically boot Windows.

[1] Used a capital letter P instead of a lower case p