



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

# Department of Computer Science

## COS132 - Imperative Programming

### Practical 1

Copyright © 2021 by Emilio Singh. All rights reserved.

## 1 Introduction

**Deadline: 26th March, 20:30**

### 1.1 Objectives and Outcomes

The purpose of this practical assignment is to provide the first introduction to some of the most important practical tasks and skills that students will need for the rest of the module. This will cover an introduction to the freeware operating system family Linux, the use of terminal commands and a command line interface (CLI), archive management and compilation of a simple C++ program using the g++ compiler. Additionally, students will be asked to investigate and familiarise themselves with some of the sites and resources available to them as part of this module.

Before continuing it is important to stress that this module, as well as many others, assumes that students will be using Linux-based operating systems in completing their tasks. Therefore, all students are strongly advised to spend time learning how to use a Linux-based operating system as soon as possible. Familiarity will generally be assumed moving forward in future practicals. Students are referred to ***Tricks of the Trade for Novice Programmers***, as stipulated in the study guide for further notes.

### 1.2 Submission

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**. The last submission made will be accepted for your mark.

### 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 Mark Distribution

Activity	Mark
Hello World	10
<b>Total</b>	<b>10</b>

## 2 The CS Website

### 2.1 Practical Sessions

The practical sessions will be conducted on Discord. Discord is a popular social media and communications app that allows many people to connect and communicate together in custom made servers. Discord allows for both text and voice communication as well as file sharing. Community management is facilitated by creators of the server and as such, Discord will enable us to effectively communicate with you throughout the course with regards to practicals. The link for Discord is: <https://discord.gg/nJe5rEqyfb>.

You will need to book a specific practical session for Discord. EBIT students are expected to make a booking for session P01 or P02. BSc and other students are expected to make a booking for session P03 or P04 or P05. Booking information will be provided in the first week of the semester. The schedule for the practical assignments can be found in the study guide.

## 3 Assignment Platform

An assignment portal is set up for you to receive and submit your practical assignments for COS 132. The portal can be found at <http://ff.cs.up.ac.za/>. Please check the portal should an announcement indicate so. Updates for practicals will be managed there. On this platform, you will submit your code for the practicals and receive your marks.

## 4 Linux

### 4.1 Overview

Linux is an open-source operating system (OS) which is part of the wider Unix family. The use of a Linux-based operating system is **mandatory** for COS 132 (and many other CS modules). You are free to use any distribution of Linux on your own computers. The facilities in the Informatorium will have their own distribution of Linux (Ubuntu).

Linux distributions can be acquired from [//ftp.cs.up.ac.za/](http://ftp.cs.up.ac.za/). This will direct you to the CS Department's file transfer protocol (FTP) server where several distributions are available. Students are of course free to use other distributions acquired from other sources.

First time users of Linux are advised to use a distribution like Ubuntu for easier initial use and setup.

## 4.2 Full Installation and VMs

Linux can be installed onto a computer like any other operating system. If something like Windows is already installed, it is possible to dual-boot, that is run Linux as a second OS, alongside existing ones. This is the recommended option for students who wish to do their practicals at home.

The alternative approach is the use of Virtual Machines (VM). A VM is an emulation of a machine that runs like any other application. In this way, it can replicate the OS environment of Linux without requiring a full installation. A program like VirtualBox is recommended for this approach. Students are referred to ***Tricks of the Trade for Novice Programmers- Chapter 4***, for further guidelines on installing Linux on your personal computers.

## 5 Archives

Archives are simply file containers that can also hold directory structures. More often, this property is combined with file compression to create compact file structures to transfer files. There are a number of types such as 7zip, rar, tar and zip to name a few. All practical assignments will come in archives that need to be extracted in order to use the files contained within. You are also required to make submissions of practical work in the form of archives. Refer to the book **Tricks of the Trade for Novice Programmers** in **Chapter 5 - Operating System Utilities** for more information on archives and compression.

### 5.1 Extracting an Archive

An archive is extracted with the following command:

```
tar -xzf XXXX.tgz
```

where XXXX.tar is the name of the new tar archive. A .tgz file is a .tar archive with additional compression.

### 5.2 Creating an Archive

An archive is created with the following terminal command:

```
tar -cvz <the list of files to archive, separated by spaces> -f XXXX.tar
```

where XXXX.tar is the name of the new tar archive. You need to have the terminal open in the file directory where the files you wish to archive are located as well.

An example of this is: **tar -cvz myFile.txt -f myTar.tar**

Only **.tar**, **.zip** or **.tgz** files will be accepted. Submissions in other formats may or may be rejected and students are advised to stick with the accepted archive formats.

## 6 Code Editing


To write programs in C++, a text editor program is needed. Several editors exist, such as Sublime Text, SciTE and others. It is advised that you try a variety of text editors and choose one that you will be most comfortable. SciTE is the editor most commonly installed in the labs.

Once you open a file in the text editor, you will then be able to write, edit and modify code that can then be compiled and run.

Opening a file in SciTE can be accomplished with both the terminal and through a desktop interface.

To open a file, **file.txt** with SciTE via the terminal type the following:  
**scite file.txt &**

Note that you need to have the terminal open in the folder where **file.txt** is located to do so.

Alternatively, SciTE can be opened by double clicking the icon, , on the desktop. Then you can simply drag **file.txt** onto the open program to open it in SciTE.

## 7 Compilation

Compilation is the process of converting human readable code, such as that written using a text editor, into a binary format for computers to understand. Compilation is an operating system specific task however, and code compiled in Windows will not work for Linux and vice versa.

For the purposes of COS 132, we will be using the GNU GCC compiler in a Linux environment. At this point, whether it is a 32 or 64 bit version is immaterial.

The steps involved in compiling a program are detailed below for a simple example program named Test.

- Compile the code by using the command: **g++ -c Test.cpp -o Test.o**  
This will convert the file into a binary code file, as Test.o.
- Create an executable program: **g++ -o Test Test.o** which will convert the binary .o file into an executable file named Test
- Execute the program using: **./Test**

Refer to *Tricks of the Trade for Novice Programmers- Chapter 6.2*, for further details.

## 7.1 Makefiles

Makefiles are special files that help automate the task of compiling and running programs. They enable relatively large programs to be compiled and run with only a few custom commands that are generally included in a makefile for convenience.

Refer to *Tricks of the Trade for Novice Programmers- Chapter 6.3*, for further notes.

## 8 Fitchfork

Practical assignments will be marked by use of an automated marking system, Fitchfork, which was developed in-house by the CS Department of the University of Pretoria. It will receive your practical assignments, as archives, and then compile and run them. This will be checked against tests from which your final mark for that assignment will be determined. All of the work related to practicals (submission and marking) is handled by at the following page: <http://ff.cs.up.ac.za/>

However, Fitchfork requires strict operating conditions so file names must be exact; that is the correct case and name, and the correct program structure. Instructions will always be provided for the correct file names and program structure. They must be adhered to otherwise no marks will be awarded.

Finally, in the event that problems with your code exist, Fitchfork will provide some feedback about the potential problems.

You are strongly advised to **test and debug** your code thoroughly before submitting it to Fitchfork. Refer to *Tricks of the Trade for Novice Programmers- Chapter 7*, for further information about testing and debugging. Each assignment will only have a limited number of upload attempts, only the last of which will be taken for your official mark, and using Fitchfork to test your code will generally waste those uploads. In the event of a compilation failure, Fitchfork will return a mark of -1. Error messages will generally be displayed in the event of failures. These messages should inform your debugging process.

As always, **no late submissions will be accepted** therefore students are advised to upload their code early to avoid potential problems.

## 9 Practical Activities

### 9.1 Activity 1

You are required to do the following:

1. Download the COS132\_Prac1.tgz archive from the COS 132 assignment page, [ff.cs.up.ac.za](http://ff.cs.up.ac.za)

2. Extract the archive to a folder
3. Verify that the archive contains two files: **hello.cpp** and **makefile**.

Since this is the first practical, nothing complex will be required. You merely need to change the comment in **hello.cpp** to output your student number plus a message. The format of this message is as follows:

```
Hello, world! I am uXXXXXXXX
```

When you open **hello.cpp** in a text editor, you will need to write the corresponding message in the area demarcated for your code. Once you are done with the edit, save the code using the hotkey “CTRL+S” (hold down Ctrl while hitting S) or by clicking on the File option the top and then clicking save. When you run the program, it should print “Hello, world! I am uXXXXXXXX” to the terminal where XXXXXXXXX is your student number.

Once this is finished, perform the following steps:

- Compile the code by using the command: **g++ -c hello.cpp -o main.o**  
This will convert the file into a binary code file, as main.o.
- Create an executable program: **g++ -o main main.o** which will convert the binary .o file into an executable file named main
- Execute the program using: **./main**
- Clear the directory using: **make clean**
- Recompile the code using: **make**
- Finally, execute it again using: **make run**

Finally, create an archive of the **hello.cpp** and **makefile** and upload this to the Practical 1 upload slot on the COS 132 assignment page. You will have 10 upload attempts to complete this activity.