|  |  |
| --- | --- |
| **01286120** | **Elementary Systems Programming** |

# Rust startup!

# Laboratory exercise 1

Rust is a new addition to a large family of high level programming languages designed to make it easy to make computers (which now include your phone and the many programmable devices that we meet everyday now!) easy to control and perform useful tasks correctly! There will be more about it antecedents and the general history of programming languages in the lectures.

For this first laboratory exercise, we will make sure that you can write, compile and execute a simple Rust program on the lab computer. You should also be able to create a simple Rust project and execute it. Optionally, you can download the Rust tools onto your laptop and check that you have installed it correctly.

## Disclaimer

Rust is a new programming language for all of us (including your lecturer and the lab tutors), so we will be learning with you. However, most of us have had experience with a wide variety of programming languages, which preceded Rust and from which many features of Rust have been derived.

## Basics

Rust programs are written in text files, using Latin characters (a-z A-Z) and a set of special characters (; , . { } ( ) *etc*). You can create a Rust program with almost any text editor, so you can start with any one with which you are already familiar. However, since you are taking a parallel course in Python, it may be more convenient to use the same editor to create Rust and Python programs, so our expert tutor will give you a brief introduction to the Python editor at the start of this lab.

The simplest way to create Rust programs is to open a Windows command line and simply type commands. (If you are using Linux, you are already familiar with this style – just continue as before 😊)

### Organizing your work

It is strongly recommended that you create a new directory for your Rust programs, using any convenient name, e.g.

mkdir RustPrograms

chdir RustPrograms

### Checking that Rust is installed

From your command line, just type

rustc

to invoke the Rust compiler. If Rust is correctly installed, you will get a page of help, describing the options that can be added to rustc.

### Building a test program

If you follow the steps in the online version

<https://doc.rust-lang.org/stable/book/ch01-02-hello-world.html>

# of Klabnik and Nichols’ text, [*The Rust Programming Language*](https://doc.rust-lang.org/stable/book/title-page.html#the-rust-programming-language)*,*

you will see instructions to make the classic first “Hello World” program.

You can cut and paste the text of this starter program

fn main() {

println!("Hello, world!");

}

from the web pages (or here!) to your chosen editor and save the program in a file with the .rs extension – the example suggests ‘main.rs’, but *xxx.rs* will work also 😊.

After you have successfully compiled and executed your first program, note that Rust encourages you to make ‘projects’ to contain Rust programs, using ‘cargo’.

Continue to the web instructions on the next page, create a new project, add a program to it and execute it.

Check that you can modify your first program in some way, e.g. change the message printed out from the boring ‘Hello World’ to something more imaginative or add more println statements or …

You can also continue on to read more from Klabnik and Nichols’ text (if the chatty style doesn’t annoy you too much) or start to look at some other, more formal, texts. Some references are available in the resources pages.

## Report for this lab

Please fill out the survey. It will, among other things, let us know what experience you have with other programming languages – and perhaps stop the lecturer from sending you to sleep in class. The English saying is ‘Don’t teach your grandmother to suck eggs!’, *i.e.* don’t waste time teaching someone things they already know.

You will also see many questions about your attitudes to online classes. We are trying to prepare for a new wave of COVID, which may send all classes back on line again.

## Forward planning for this course

Assuming there is no significant change in the COVID (or MonkeyPox or …) situation it is currently planned

(a) to have **ALL** of the lab classes on campus in this lab and

(b) to have ***most*** of the lecture classes on campus also.

However, (b) is flexible and, depending on your feedback, some of the lectures may be presented on line. Indicate clearly in the survey if you do **NOT** have access to a suitable computer or bandwidth at home or at some other place off campus. As long as COVID does not cause another complete lockdown, you will be able to follow online lectures on campus also – either in the lecture room or this lab. The lecture slides will also be available in the resources section.

## Use of data

***Answers to the survey questions are entirely voluntary***. One KMITL research student is studying attitudes to online classes and may extract statistical data and any anonymous comments you choose to add. The survey will be separated from the top page. Your name will only be recorded *for lab attendance purposes only* and your answers to the survey will be separated and anonymous. However your answers may help planning of this course, so please spend the time to fill in the questions and add any comments of your own.

If you are interested, or just curious, any published papers using this data will be available: ask towards the end of the semester – just ask.