

Introduction

Since many of you have not programmed before and you are not familiar with the VG101 course, it is necessary for me to give some brief introductions.

What do we learn in VG101?

We all know that we will be learning programming in this course. But it is more than just programming. It will focus a lot on programming like an engineer, which means that you will cultivate your skills of organizing big programming projects in VG101 and you will have many chances to cooperate with others.

So if you have any friends in other institutes learning (or having learned) programming in C or C++, you will find that what you are learning are somewhat different. And as a result, the reference you can get from them is very limited.

But don't worry, most of the things in this course are simple and clear enough for you, just follow the progression of the classes and your assignments, and you will get a good grade.

About the languages

Matlab

Different from other schools or institutes, we start our programming class from Matlab. Matlab is a strong language in mathematical calculation, and in fact the name is the abbreviation of "Matrix Laboratory". All things in Matlab can be regarded as matrixes. And based on the various operation on matrices, Matlab can do things from being a strong calculator in your daily life to training machine learning systems, as many of machine learning systems are based on matrices. Another thing is that Matlab is good at plotting data and image processing, while it will take you a lot of effort in C and C++.

Since Matlab is the first part of your VG101 learning, you will also meet with some basic ideas about programming. An important word for you is "algorithm", which means the method to solve a problem. You will learn some simple algorithms, among which the most important is "recursion" in my opinion. I will spend much time in this tutorial to get it clear for you about recursion, since many of you might get confused about the concept.

The things you should pay attention to in Matlab part: plotting, designing algorithms(especially recursion) and exploring the rich function library of Matlab(It has a documentation system to help you get familiar with the functions and it can even be accessed offline(which is valid in exams))

C

Compared with Matlab, C and C++ are more close to the working mechanism of computers. So although Matlab is strong, but it will run much slower. In the beginning of learning C, you will learn some basic operations, and I think the most important part is inputting data and outputting data, as different inputting and outputting functions might have different features and limitations.

What's more, in C you will learn a new type of data called pointer. A pointer contains an address of a data saving unit in the computer. So C language will help us get a better control of the internal storing system in the computer, which will allow us to do many things which we cannot accomplish in Matlab.

However, the programming process in C will be harder than in Matlab. And as the pointers will operate directly on the internal storage of computer, the programs will be more likely to collapse. So be careful while programming in C.

The things you should pay attention to in C part: algorithm designing, inputting and outputting(also in files), the use of pointers

C++

As for the basic operations, C++ is much like C(in fact, a program written in C language can also run in C++). There mainly two features of C++.

The first is called "classes", and it is the core of "object oriented programming"(OOP). Maybe some of you have heard this before, but you don't know what does this terminology mean. And I will pay more attention on explaining things relative with classes for you in the future.

Another is called "external libraries". In this part, you will learn to use a new library(you can regard it as a plugin for C++) called OpenGL, and this library will help you to plot and draw pictures using C++.

The things you should focus on in C++ part are just the two things above.

README file writing

In this course you not only need to program, you should learn how to write a descriptive file for your programs as well, and this file is always called "README". For each assignment and exam, you should write one README file.

Here are things that you should contain in the README file

- Brief instruction of running the programs (this can be same in every README file if you are using the same language)
- Description about what your code can do and the algorithm you use(for every program)
- Sample input and output for every program.
- Change log of your assignment if you have had revised it several times.
- Some bugs that you cannot solve yet or some limitations of your codes.
- How to contact you(usually your email)

You can just write README file in a text editor(The final name should be README.txt), but I also want to recommend you a useful tool to write README files which is called markdown. It is very simple enough for you and it will make your README much more neat.(In fact, the tutorial files in this repository are all written in markdown). If you want to use markdown, the file name should be README.md.

In fact, what you can use to write README files are just the two mentioned above (text editor and markdown). Don't use word or any other document editor to do this.

README files are more important than what you can imagine. It will guarantee you a good score in assignments and will win you many partial points in exams, as the exams are usually very difficult.

Commenting on your code

A good habit of programming is writing comment. In different languages, there are different ways to write small comments and notes just beside your code. If you think your algorithm is complex, you need to add some comment to explain in some crucial steps. And if you think there are something still need to be improved, you can add comments to remind you. Also, comment will sometimes win you some partial scores, but its possibility is much lower than a good README file.