ENDG233 Programming with Data

Week 1 - Course Introduction

ALS1 September 6-9

First, a huge welcome to the Schulich School of Engineering and to one of your first year courses, ENDG233! You are making a big step from the familiar high school environment to the university environment with an exciting fall term ahead of you. This document contains a lot of information regarding ENDG233 and how to get started with the course. Your fist weeks assignment will be getting access to the on-line textbook for the course, setting up a Python interpreter on your computer, finding the content of ENDG233 on D2L and going through the course content, and reading through this rather lengthy introduction. In your first session of ENDG233 this week we will go over these details and help get you set up.

What is the course about?

Let's start with the course description that you will find in the University of Calgary calendar for this course: University of Calgary : Digital Engineering ENDG (ucalgary.ca)

Fundamental programming constructs and data structures. Algorithm development and problem solving. Programming techniques to facilitate data analysis. Obtaining and cleaning data. Data validation. Data manipulation. Data visualization. Introduction to decision making using machine learning. Applications chosen from all engineering disciplines.

So that is a bit of a mouthful and rather bewildering. OK so let's break it down.

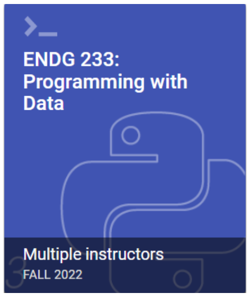
The overall primary learning objective is straight forward. Essentially, we will be teaching the ***foundations of the Python programming language*** with a hands-on practical approach to ***how to write a computer program to do something***. If you have no programming experience that is not a problem. We will go from ground zero. At the end of the term you will know how to write a basic program for a given application. You will learn how to write expressions and tie this together in a script file including functions, modules and file storage. With this introduction you will be able to write an effective program for a wide variety of applications. If you have prior Python programming experience you will find this course will fill in many gaps in your knowledge of Python and will provide many opportunities for developing good programming prowess and style. Regardless of background, at the completion of ENDG233 you will have a solid platform from which you can launch into more advanced programming and application of programming to real engineering problems whether you are in software, civil, mechanical etc. Programming is emerging as that indispensable tool for doing anything-engineering design or analysis type calculations.

Through this course you will be introduced to manipulating and processing data in a computer. We will introduce basic plotting and data visualization in Python. We will also be learning about efficient ways of storing the data in arrays and matrices and how to manipulate the data. Will we get into machine learning or other engineering applications? Clearly it is limited what we can cover in the next four months as we are starting from the very beginning. To say that you will be able to write programs for machine learning and computational engineering is a stretch. However, we will get to the point that once you learn a bit more math and numerical processing, then armed with the Python that you will learn in this course, you will be ready to tackle that.

How is the content of the course organized?

ENDG233 is a flipped class, a teaching methodology you have been introduced to in your earlier orientation. This means that we will not be holding formal lectures that you all have to attend. Instead, we have placed the lecture content on line in the form of videos that you can watch outside of class hours. As well we have selected an on-line interactive textbook for this course that you will have access to for the duration of the course. More details on this later. This frees up the weekly class for 'active learning'. In ENDG233 you will have one active learning session (ALS) per week that will span a three hour period. During the ALS you will be working on your assigned programming problems and questions. The learning is interactive in that the instructor and the teaching assistants (TA's) will be there to help you. Simply put, the more you participate in the ALS the faster and better you will learn to master the Python programming language. When you come to the ALS be sure that your PC is ready with an installed Python interpreter. There are various options for the Python interpreter and the TA's can help you select the option that is best for you. Please see the note on the Downloading and Installing of the Python interpreter and environment that is in the D2L content area for this course.

You will need to get the course textbook which is an online interactive guide to learning the fundamentals of the Python language.



Details will follow on how to purchase the book from the UC Bookstore. Going forward, we will refer to the textbook as the 'Zybook'. Note that your purchase of the zybook gives you personal access until January 2023. However, you can download a PDF of the book, it just will no longer be interactive.

The fall semester consists of 14 weeks between now and when the final exams start. One of those weeks is the fall reading week in November leaving 13 weeks. the organization of the course content in this 13 weeks is listed in the table below. Note that 'ALSn' implies the n^{th} ALS for week n. The content is detailed the Python related terms are unfamiliar. Don’t worry as this will become clear as we progress through the course. The point is that every week there is a fair bit to cover and learn. The course also build on the material covered as you would expect in a course on programming.

ALSn	date	ALS details
1	Sept 6-9	Intro to the course, set up Zybook and the Python environment, visit D2L site, meet the instructor and TA’s. Get started on the first few sections of chapter 1. **assignment:** Read this document, Ch1 of Zybook, answer questions
2	Sept 12-16	Introduction to programming, algorithms and Python **assignment:** Ch1, Ch2, D2L videos, Questions and basic Python program with input(), print()
3	Sept 19-23	Conditional if-else branches, indentation, Python coding style and convention. **assignment:** Ch2&3, Questions and some program examples, videos on branching logic parts 1,2,3,4
4	Sept 26-Oct 1	data types strings, lists, tuples, dictionaries **assignment:** Ch4, Questions and some program examples, videos on compound data types parts 1,2,3,4
5	Oct 3-7	**learning objective:** loops, breaks, user defined functions, passing parameters, errors, namespaces, scope resolution docstrings, **assignment:** Ch5 and Ch 6 intro, videos on repetition structures parts 1,2,3,4
6	Oct 10-14	**learning objective:** functions and namespaces, user defined, passing parameters, errors, namespaces, scope resolution docstrings, engineering examples **assignment:** Ch6 and videos on functions 1,2,3,4, Questions and some program problems
7	Oct 17-21	**learning objective:** Advanced Strings, Lists, dictionaries, part 1 **assignment:** Ch7 working with strings, lists and dictionaries, sorting lists videos on advanced strings, lists, dictionaries parts 1,2,3,4,5, Questions and program examples
8	Oct 24-28	**learning objective:** Advanced Strings, Lists, dictionaries, part 2 **assignment:** Ch7, videos parts 6,7,8, Questions and program examples
	Oct 24	**Midterm I**
9	Oct 31-Nov 4	**learning objective:** Modules and packages finding and installing modules, importing, packages,math,plotting, numpy,matplotlib **assignment:** Ch8, module videos part 1,2,3,4, Questions and some program examples with built in modules
10	Nov 7-11	Reading week no new content
11	Nov 14-18	Working with numpy arrays and matrices, matplotlib.pyplot **assignment**: Read supplemental material and Zybook section 25.4 Questions and
	Nov 22	**Midterm II**
12	Nov 21-25	data plotting and visualization **assignment:** Questions and some program examples
13	Nov 28-Dec 2	File reading and writing **assignment:** read Ch10. Questions and some program examples, programming
14	Dec 5-7	Summary and preparation for final (no active learning assessment)

		assignment: Review programs and overall summary questions, prepare for final

*There are 12 active learning assessments. Weeks 10 (reading week) and 14 (final exam preparation) do not have active learning assessment.

**Classes end on a Wednesday. Students in a Thursday section can join either on a Tuesday or Wednesday section to prepare for the final exam.

So the flow of the course is to begin with a general introduction to computers and programming. Then we introduce Python and start going through the basic components and data types of the language. Then we move into numpy and pyplot which are standard Python libraries for numerical array processing and plotting.

Note that there is an assignment associated with every ALS, as listed in the table. You can only learn python programming by writing a lot of programs to do various things. Hence in this course you will have assigned programs to write every ALS. The assignment of programming problems will be posted on D2L the Friday before the upcoming ALS. During the ALS you will be working on the programming assignments with the help of the instructor and TA's. Collaboration with your other classmates and friend is encouraged and the ALS is great opportunity to do this. The objective is to finish the programming assignment within the ALS period. However, if you need more time then you can continue to work on it outside of the ALS.

When you are done, you can choose to go through an individual assessment of your work. This can be done towards the end of the ALS by one of the TA's. However, do to the congestion of getting the assessments done, you can delay the assessment until the following week. This can also give you some extra time to thoroughly understand your code.

The assessments are done by the TA and on an individual one on one basis. Hence collaborate as much as necessary with friends and classmates to get the assignment completed. However during the assessment, you will have to demonstrate your programs and describe how they operate to the TA. More on this later.

On to some course admin details as contained in the course outline.

How is my grade for ENDG233 going to be determined?

There are several components that make up your overall grade that are listed in the table below which also gives the weighting of each component:

Component	Weight
ALS student assessment	16%
Midterm #01	22%

Midterm #02	22%
Final exam	35%
Engineering Attributes Reflections	5%
Total	100%

The ALS assessments, that were briefly discussed above, are weighted at 16%. Then we will have two midterms in this course that each have a weighting of 22%. Then there is a final exam that is worth 35%. The last 5% is based on your professionalism course that is outside of this course. That is we basically receive your grade for this component and just calculate it to determine your overall grade.

What will the midterms and final be like?

The midterms and finals will be multiple choice exams. Both midterms will be 2 hours in duration and final will be 3 hours. All three exams will be in-person and all sections of ENDG233 will write the exam at the same time. The schedule for the midterms is:

Midterm 1: 2 hrs from 7-9pm on Tuesday Oct 26

Midterm 2: 2 hrs from 7-9pm on Tuesday Nov 22

Final exam: date and time to be disclosed

For all three exams you are allowed a single two-sided aid sheet. There is no restriction as to the print type nor content of the aid sheet. A suggestion is to not put so much detail on your aid sheet that you cannot read it during the exam! No calculator, laptop computer, tablet, personal digital assistant, cellular phone, or other electronic devices will be permitted during the midterm and final exam.

If you cannot make it for a midterm exam due to circumstance (implying illness, family emergency, accident etc.) then the default is that you get zero for that exam. There will not be a makeup exam for either midterm. However, if your ‘circumstance’ is legitimate then you can ask your instructor for a redistribution of your marks. That means increasing the weighting of the next exam and final to compensate for the missed exam. For this redistribution we have the following rules:

1. A student that misses midterm 1 or midterm 2 due to circumstance, needs to inform the section instructor no later than 24 hours after the start time of the scheduled exam.

2. Depending on the ‘circumstance’, the instructor has the option of granting redistributing the students grade based on the process described below. The term ‘qualifying for grade redistribution’ implies the instructors discretion in regards to the ‘circumstance’.

Missing midterm 1 and qualifying for grade redistribution the 22% allocated to midterm 1 will be redistributed such that the grade allocated to midterm 2 is increased to 33% and the grade allocated to the final is increased to 46%.

Missing midterm 2 and qualifying for grade redistribution the 22% allocated to midterm 2 will be redistributed such that the grade allocated to the final is increased to 57%.

Missing both midterm 1 and midterm 2 but qualifying for grade redistribution, the 44% allocated to midterms 1and 2 will be redistributed such that the grade allocated of the final is increased to 79%.

Note that if due to circumstances that you cannot write the final exam then discuss this with someone at the ESC. They will let you know what options you have. You may qualify for a deferred exam which is held in January instead of the usual exam time of mid-December.

More on the ALS assignment student assessment

For a given ALS there will be an assignment consisting of several programming problems to complete and some questions. You can have an assessment done on your completed assignment up until the end of the following week's ALS. The TA's will be quite busy doing these assessments so that you may request a time on a form that the instructor will provide. More details will be given within the ALS. The TA doing your assessment, will want to see the code that you have written in response to the coding questions. Then the TA will ask you to demonstrate one or two of the programs. Show that the program meets the design objectives and describe the workings of the code to the TA. The TA can then ask several questions related to your code or some of the questions in the assignment. The assessment process will take about 5 minutes to complete. Upon completion you will be given a score in the range of 0 to 5 based on the following criteria.

mark	criteria
0	problems not complete, programs not working and/or not written by student, student can't explain how programs work
1	programs/problems barely done, can't be demonstrated and/or explained
2-4	variable degree between 2 and 4, programs partially work, partial demonstration, problem questions partially answered correctly.
5	student explains and demonstrates programs well, answers all questions regarding programming problems and assignment questions to the TA's satisfaction

Hopefully, you will agree with the TA regarding the grade of the assessment.   If not, while there is no provision for redoing the current assessment, there is an opportunity to do better in the next assessment. If the TA gives less than 5 as the score then an explanation will be provided as to where you faulted and where you can improve for next time. Remember also that we only **count the best 8 assessments** for calculation of this grade component. During the term you can accumulate up to 12 assessments.

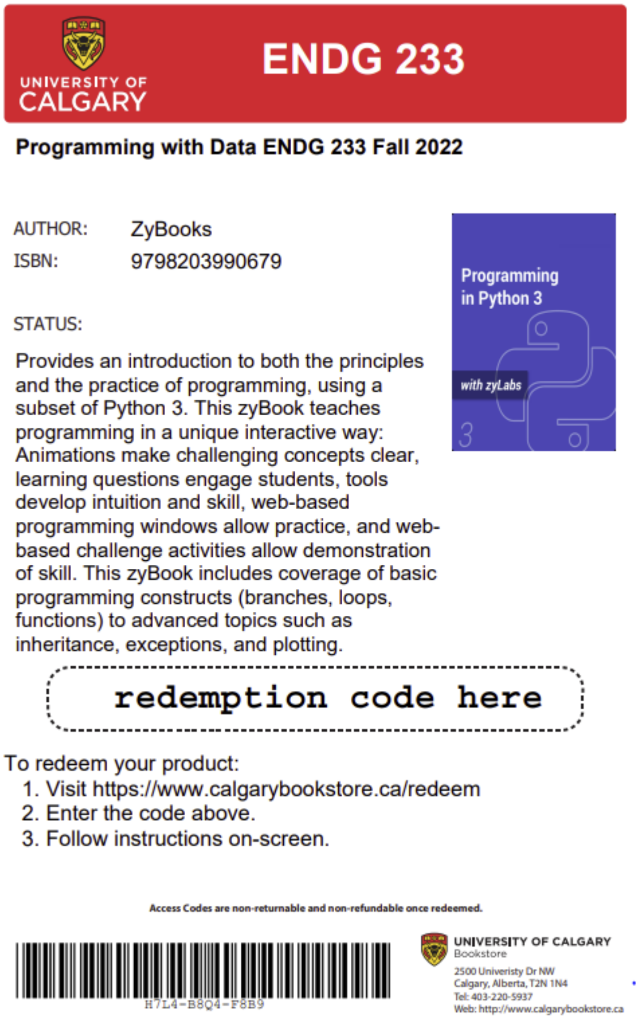
How Do I get the Zybook ordered and setup?

The direct link to the ZYbook for ENDG233 at the UC Bookstore is:

https://www.campusebookstore.com/integration/AccessCodes/default.aspx?permalinkId=05e74243-ac3f-40f7-9460-e5c1b0119090&frame=YES&t=permalink.

Purchasing online is the fastest as you will have an access code and instructions emailed almost immediately to your account after purchasing. The email will consist of a link to set up your Zybook account and access the on-line Zybook.

However, if you would rather go to the bookstore, there will be cards on the shelf in the ENDG section of the bookstore that you can take to a cashier to have activated. The card looks like the following:



UNIVERSITY OF CALGARY

ENDG 233

Programming with Data ENDG 233 Fall 2022

AUTHOR: ZyBooks

ISBN: 9798203990679

STATUS:

Provides an introduction to both the principles and the practice of programming, using a subset of Python 3. This zyBook teaches programming in a unique interactive way: Animations make challenging concepts clear, learning questions engage students, tools develop intuition and skill, web-based programming windows allow practice, and web-based challenge activities allow demonstration of skill. This zyBook includes coverage of basic programming constructs (branches, loops, functions) to advanced topics such as inheritance, exceptions, and plotting.

redemption code here

To redeem your product:

1. Visit https://www.calgarybookstore.ca/redeem
2. Enter the code above.
3. Follow instructions on-screen.

Access Codes are non-returnable and non-refundable once redeemed.

H7L4-B8Q4-F8B9

UNIVERSITY OF CALGARY
Bookstore
2500 Univeristy Dr NW
Calgary, Alberta, T2N 1N4
Tel: 403-220-5937
Web: http://www.calgarybookstore.ca

After redeeming that card, you will be emailed with the link to activate your Zybook account and access the online Zybook.

Getting started…

So enough detail on the admin stuff. Let's now finally get started on the content! Start by going through the sections 1.1 to 1.5 of the Zybook.

Along the way you will have read a couple of snippets regarding Python and this is really where our journey begins.

The root of Python goes back to the 1980s when a software programmer, Guido van Rossum, became bored during a Christmas break. As a programmer, he was well aware of the limitations of existing programming languages (notably C, Pascal, Fortran, Cobal etc.). He came up with a novel way of structuring and processing data that led to the humble beginnings of Python. Initial uptake and interest in Python was slow but grew steadily. It took until 1994 to get Python 1.0 released as a usable computer language. Being open source such that any programmer could contribute to the evolution of the language it quickly took off thereafter. This accumulated contribution from many sources led to Python emerging as a powerful very well designed computer language. We can say that Python's success is a result of the many programmers who adopted it and contributed towards improving it and providing the many libraries that we currently have access to. Currently, 13.5% of software developed worldwide being written in Python which makes it the most popular programming language used. Through the introduction of Python in ENDG233, and learning of the data forms and language structure, you will begin to appreciate why it has become so popular. Python3 is the latest release of Python that occurred in 2008. This course is based on Python 3. Variants are continuously being released with 3.7.9 considered stable for general use.

The reason for the popularity is that it is an easy interpretive language to learn that is structured in a way that makes the syntax concise and powerful. Python code looks like geekish English sentences and is logical and systematic. The data structures allow for sophisticated manipulation of all forms of data whether it be numbers or character strings. Where the base language is deficient, there are extensive libraries that are coupled into the code. But it is the community of active python programmers willing to share code, and constantly looking for improvements, that is the key enabler for Pythons continued success and emerging dominance.

Python is an interpretive language that is dynamically typed. We will shortly learn what that means. However, while the Python language makes for concise powerful code that is relatively easily written and read, it is this factor that also makes it not efficient in terms of memory or CPU processing. Fortran, C and C++, on the other hand, being strongly typed and compiled to run faster, can use CPU MIPS and memory more efficiently. Efficiency is required for working with processing intensive big data applications. However, this is where pre-compiled library functions come in that you couple together with your regular Python code. This significantly speeds up the code execution. Also, computers nowadays generally have overcapacity in memory and CPU performance in typical applications that are run. The slight speed disadvantage of Python+libraries is therefore typically insignificant. Specifically in the course, you will be introduced to the library of numpy which has very efficient numerical array processing structures. With numpy and a reasonably fast computer, you will find Python can handle most data processing problems sufficiently well.

Questions to consider:

1. Why do you think a pre-compiled language such as C++ can generally run a data processing requirement faster than Python?

2. Consider the student assessment of an ALS assignment by the TA as a program. Write a flow chart that the TA could follow to come up with your assessment grade of an integer of the set {0,1,2,3,4,5}. The input would be your programs and answers to TA questions.

3. Based on the flow chart is #2 think about how the ALS assessment can be done with an AI program and some robotic platform.   What would be the difficult flow chart steps to implement in the AI program?