Lecture 1.1 : Module overview

Introduction

- CA117 is a second course in computer programming using the Python language.
- It is assumed you are already familiar with the Python programming environment (running the Python interpreter and editing text files), basic Python data types (strings, lists, dictionaries, etc.) and are able to write short programs that make decisions with `if` statements, iterate with `while` and `for` loops and that use functions to decompose a problem.
- This course will enhance your programming skills so that you will be able to write more complex programs using more sophisticated techniques, in particular those of object-oriented programming.
- It is a practical programming course that requires a commitment to writing lots of programs.

Module homepage

- On the homepage you will find lecture notes, lab exercises, bucketlist assignments, announcements, links to videos, useful links, further reading, etc.

Einstein

- As with CA116, upload your programs to Einstein to have them verified.
- As with CA116, Einstein will track for you your progress (in terms of completed lab exercises) compared with other students in the class. (Student details apart from your own are anonymized.) Use this facility to track how you are doing compared to your classmates. Obviously, your aim should be to avoid falling too far behind.

Timetable (weeks 1-12)

- Tuesday:
    - 0900-1100 : lecture in QG13
    - 1400-1600 : labs in L101, L125, L128, L114
- Thursday:
    - 1100-1300 : lecture in QG13
    - 1400-1600 : labs in L101, L125, L128, L114
- Timetabling is not under my control so if you have any issues with the timetable raise them with the chair of your degree programme.

Lectures

- Normally each lecture will:

1. Introduce some Python programming concept, and subsequently,
2. Demonstrate the practical application of that concept in code.

- Normally a lecture involves me coding in a Python notebook. These notebooks are made available at the start of each week. I recommend you take a copy and follow along with me on your laptop during a lecture.
- Normally some time is allocated each week to reviewing previous lab exercises. During these sessions we will examine some common programming errors extracted from code uploaded by anonymized CA117 students.
- 2021's lectures were delivered on-line. These will be made available to you.
- Additional videos will also be posted covering lecture material, walkthroughs of solutions to selected lab exercises, programming tips, pointers to on-line resources, etc.

Continuous assessment

- Continuous assessment accounts for 50% of your overall mark in this module and is made up of four components:

CA component	Marks	Due
Labs	5%	Weekly
Bucketlist	10%	Weeks 7, 12
Lab exam 1	15%	Week 6 (Tuesday 15 February 1400-1550)
Lab exam 2	20%	Week 12 (Tuesday 29 March 1400-1550)

- The program below calculates your final CA mark (as an overall percentage) from your marks in each of the four CA elements. Substitute your own marks to work out your final CA mark.

```
#!/usr/bin/env python3

from decimal import Decimal, ROUND_HALF_UP

def main():

    # Sample marks in each CA component
    labs = 66
    bucket_list = 50
    labexam_01 = 33
    labexam_02 = 44

    ca = (1 * labs) + (2 * bucket_list) + (3 * labexam_01) + (4 * labexam_02)
    ca = ca / 10

    # Round to nearest integer (with .5 always rounding up)
    ca = int(Decimal(ca).to_integral_value(ROUND_HALF_UP))

    print(f'Your overall CA mark is: {ca:d}%')

if __name__ == "__main__":
    main()
```

```
Your overall CA mark is: 44%
```

Labs (5%)

- You cannot learn how to program from lecture notes alone. Programming ability is a *skill* that is acquired through practice and through learning from mistakes.
- You must *at a minimum* complete all lab programming exercises. Do not expect to always complete each set of lab exercises during the scheduled lab time. You are expected to dedicate considerable additional private study time in order to master the course material.
- Similar to CA116, when on campus, lab tutors will be present to help you with programming exercises. Their role is to help you find a suitable solution and **not** to provide you with a solution.
- Similar to CA116, you are required to upload your code to Einstein. It will help you verify you have successfully solved lab exercises. (It will also help me identify common programming errors that can be addressed in class.)
- Lab exercises contribute 5% to your overall mark in this module. Each lab has an associated completion deadline. Credit is awarded only for successfully completed lab exercises that are submitted before the deadline. For example, if over the course of the semester you successfully complete 60 of 80 lab exercises on time then your lab mark would be 75%.
- To receive the marks available for a lab exercise it must pass all of the associated test cases on Einstein.
- Normally at least one of the test cases on Einstein for any particular lab exercise will be hidden. This means you will not be able to see the inputs used to test your program.
- Passing the public test cases but failing a hidden test indicates you have not sufficiently tested your code and you will need to devise your own additional test cases based on the problem description (coming up with test cases is part of the exercise and something real world programmers have to do).
- Selected solutions to previous lab exercises will appear here.
- All lab submissions must adhere to DCU's academic integrity policy. See Academic integrity.

Bucketlist (10%)

- A **bucketlist** of two programming assignments will be posted over the course of the semester. These programming assignments contribute 10% to your overall mark in this module.
- Bucketlist exercises must be uploaded to Einstein before their associated deadline.
- All bucketlist submissions must adhere to DCU's academic integrity policy. See Academic integrity.

Lab exams (35%)

- Two lab exams will take place over the course of the semester and contribute a total of 35% to your overall mark in this module.
- **Lab exams will take place during the Tuesday lab slots of weeks 6 and 12.**
- **Lab exams cannot be rescheduled.**
- The first lab exam is worth 15% of your overall mark.
- The second lab exam is worth 20% of your overall mark.
- Skills covered will be a selection of those acquired by completing all preceding lab exercises and sample lab exams.
- Each lab exam lasts 1 hour and 50 minutes.

- You must sit in your assigned lab.
- Unless otherwise stated lab exams are posted at the bottom of the CA117 home page. Lab exams become accessible at the exam start time.
- Upload your solutions to Einstein. Login here with your DCU credentials. If you fail to login here with your DCU credentials you will receive a mark of zero.
- Only code uploaded to Einstein during the lab exam will be graded. All uploads are time-stamped. The submission deadline will be strictly enforced.
- It is your responsibility to ensure your Python code is compatible with Einstein's Python version.
- All exercises must be completed (unless otherwise stated).
- Your programs may only import from the following Python modules: `sys`, `math`, `re`, `string`.
- Your programs are not permitted to use `eval`.
- During the exam you will have access to the CA117 web site.
- During the exam you will **not** have access to videos and notebooks on the CA117 Google Drive.
- During the exam you will **not** have access to model solutions.
- All lab exam submissions must adhere to DCU's academic integrity policy. See Academic integrity.
- Sample input and output are provided for each exercise. These sample test cases are not exhaustive. Passing such sample test case(s) does not guarantee marks for that exercise. Your mark for each exercise is calculated based on additional test cases applied during the marking process. See Marking of bucketlist, lab exam and final exam exercises.
- It is your responsibility to save your work regularly.
- It is your responsibility to upload your work regularly.
- If you miss the exam due to illness, **you must submit a medical cert** to cover your absence.
- A sample lab exam will typically be supplied in advance of the real lab exam.
- A sample lab exam solution will typically be supplied in advance of the real lab exam.
- The sample lab exam and the real lab exam will **not** be the same.

Lab exam FAQ

- **Q.** I cannot attend the lab exam because I am playing sport for DCU/Dublin/Leinster/Ireland/Europe etc. on that day. What can be done?
- **A.** A supplementary exam will **not** be organised to facilitate students who do not attend a lab exam. Thus you need to confirm well in advance of the lab exam and with the **chair of your degree programme** how your situation is to be handled.
- **Q.** What if I have a question during the lab exam?
- **A.** It should be clear from the exercise description and the example input and output what is required. However if you are confused you can ask me or an invigilator.
- **Q.** I very nearly solved question X. Am I awarded attempt marks?
- **A.** See Marking of bucketlist, lab exam and final exam exercises.
- **Q.** My program passed the supplied test cases but did not receive full marks. Why not?
- **A.** See Marking of bucketlist, lab exam and final exam exercises.
- **Q.** Why is use of `eval` not permitted?
- **A.** It's bad programming.
- **Q.** My computer crashed/exploded/disappeared during the exam. Can I have an extension?
- **A.** No. An extension of 15 minutes for technical difficulties has been built into the exam duration.
- **Q.** I uploaded my code a mere X minutes after the deadline. Will it be marked?
- **A.** No. An extension of 15 minutes has been built into the exam duration.

- **Q.** My program works with the version of Python I have installed at home but does not work with Einstein's version. Can I have some marks?
- **A.** No. It is your responsibility to ensure your code is compatible with Einstein's Python version.
- **Q.** How should I test my program?
- **A.** In general, you need to read the question carefully and identify the range of possible inputs. Following that, identify any *boundary* or *edge* or *awkward* cases since it is often here where problems can arise. As a simple example, if the exercise states input is an integer in the range [-100, 100] you might devise the following test cases:
    - -100 (end of range),
    - 100 (end of range),
    - 0 (positive/negative boundary),
    - -1 (start negative range),
    - 1 (start positive range),
    - -50 (mid-range negative),
    - 50 (mid-range positive).
- **Q.** Have you any lab exam tips?
- **A.** Some general tips:
    - Read each question carefully. Study the example input and output.
    - Some questions are easier than others (typically the early ones) and I would do the easier ones first.
    - Try not to panic. If a particular lab exam goes poorly you have other opportunities to catch up: labs, bucketlist, the other lab exam, the final exam.
    - Your final upload for each question is the one that counts so ensure your final upload is your best attempt.
    - Know where to find things in the notes.

Final exam (50%)

- A final exam, worth 50% of your overall mark, takes place in May.
- The exam lasts 3 hours.
- The exam is lab-based and the normal lab exam rules apply.
- All exam submissions must adhere to DCU's academic integrity policy. See Academic integrity.

Marking of bucketlist, lab exam and final exam exercises

- Test cases, where supplied on Einstein, will be visible but will **not** be exhaustive. These *sample* test cases will serve as an aid to developing your solution.
- Devising your own additional test cases is considered part of the exercise (as it is part of being a programmer in the real world).
- Passing all tests cases on Einstein will thus not guarantee all marks for that exercise. Additional tests will be run on your code afterwards as part of the marking process.
- Marks may be awarded for reasonable attempts so ensure your final upload for each exercise is your best attempt.
- Marks may be deducted for code that is incomprehensible, overly long, poorly structured, inefficient, incomplete, etc.

Academic integrity

- Ensure you read DCU's Academic Integrity and Plagiarism Policy.
- Sharing your work with anyone is a breach of the above policy.
- Copying work from anyone is a breach of the above policy.
- All Einstein submissions will be actively monitored for signs of collusion/copying.
- Any breach of the above policy is a serious offence that will result in penalties and/or the application of disciplinary procedures.

How to pass

- Your overall mark is calculated from two components: continuous assessment (worth 50%) and one written exam (worth 50%).
- To pass the module your overall mark must be 40+.

Resit information

- Only in the event that your *overall* mark is less than 40 must you resit some component(s):
    - If you failed the final exam you must resit it.
    - If you failed the CA you must resit it.
    - If you failed the final exam and the CA you must resit both.
- Resit CA consists of a number of programming exercises to be completed remotely and an on-campus lab exam (these are worth 50% of your overall resit mark).
- The resit exam (worth 50% of your overall resit mark) will take place in August 2022 and takes the form of an on-campus lab exam.

Continuous assessment extensions and absences

- As per DCU regulations and in order to be fair to all students, any request by a student for an extension to any continuous assessment component(s) must be accompanied by a medical cert.
- Absence from a lab exam due to illness must also be covered by a medical cert.

Attendance

- Attendance at lectures and labs is recorded.
- Lab attendance tracking is automated. To be marked present for lab N you must successfully complete and submit to Einstein one exercise from lab N no later than 15 minutes after lab N's scheduled end time.
- **Completing an exercise from a prior lab does not count as lab attendance.**
- Attendance is tracked only to monitor student engagement. It does not contribute to your final grade.

Resources

- All course notes and programming exercises will be made available on-line.
- Should you require further resources, there are numerous Python programming books available in the library (some are available as e-books) including the following:
    - *Starting out with Python* by Gaddis,
    - *Introducing Python* by Lubanovic,
    - *The Practice of Computing Using Python* by Punch and Enbody,
    - *Introduction to Programming in Python* by Sedgewick, Wayne and Dondero.
- There are also numerous Python programming tutorials available on-line including the following:
    - Think Python: How to Think Like a Computer Scientist,
    - A Byte of Python,
    - Learn Python the Hard Way,
    - The Python Tutorial.

Struggling with programming?

- Check out Computer Science Circles.
- Check out Khan Academy's Python YouTube Channel.
- Sign up for edX and check out their programming courses (to earn a certificate you have to pay but you can audit some courses for free).
- Sign up for Coursera and check out their programming courses (you can audit some courses for free).
- Take a look at Udemy for Python programming courses.
- Contact the School's Programming Help Desk.
- Check out the Python programming e-books available from DCU library.
- Study and attempt past exam papers.
- Use a visualizer to step through code so you can see how it works.
- You need to practise programming at home so get Python installed on your laptop or PC (see below). As you build experience code patterns will start to come to you automatically e.g. how to read in every line from `stdin`, splitting it and building a dictionary from it, etc.
- When writing a program, write it piece-by-piece and save and run it after each step to slowly build-up a solution.

Python programming off-campus

- You should install Python 3 on your laptop or PC so that you can practise programming while off-campus.
- Alternatively log into TermCast (if you use TermCast during labs I can drop in to your session to offer advice).
- Google Colab comes with a built-in Python interpreter so you could also use that.

Contacting me

- Talk to me during a lab or lecture.
- E-mail me from your DCU e-mail account with CA117 in the subject.

- An anonymous survey is normally distributed in week 5 and provides you with an opportunity to supply feedback on how the module is working for you and on how you think it might be improved.

Slack

- There is a CA117 Slack channel.
- Use it to ask questions and answer your classmates' questions.
- Go to the sign-up page and join the "#ca117-general" channel.

Python version

- We will be programming on Linux with Python 3 (your code must be compatible with Python version 3.9.2 running on Einstein). To invoke this version of the Python interpreter you must enter `python3` at the command prompt (and not `python` which runs Python version 2):

```
$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello')
hello
```

- The Python 3 standard library documentation is a resource you should consult regularly.
- Your code should adhere to the Python Style Guide.

Module synopsis

- We will begin by reviewing, consolidating and, where applicable, extending the programming skills you acquired through completing CA116.
- We will then move on to study *object-oriented programming*.