bianca.phelan@tudublin.ie

O Object
O Oriented
P Programming

S1 = Python with Bianca

S2 = Java with Bryan

DT228(TU856)/DT282(TU858) - 2

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH

DUBLIN

TECHNOLOGICAL UNIVERSITY DUBLIN

COMPUTER SCIENCE

Lecture Week 1
Introduction Part 1

Learning Objectives

- Describe the aim, structure and assessment strategy of this module
- Know your lecturer
- Intro to OOP and Python

Course Organisation

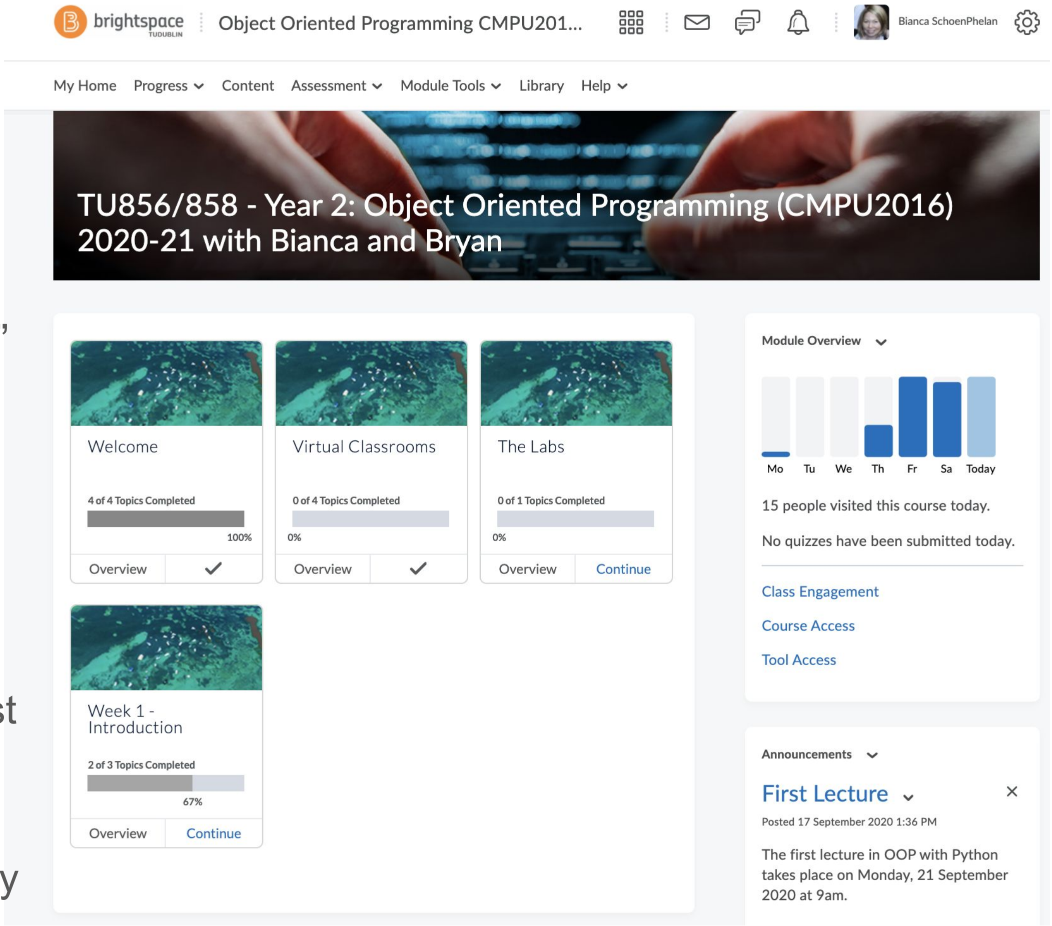
Please check timetables and brightspace announcements for class and rooms changes.

- All materials and announcements through **brightspace**
  - Please be aware that DT211C have a different OOP lecture
  - Students self-enrol
- Lectures on **Mondays, 9-10am** and **Tuesdays 3-4pm**
- Tutorials on **Wednesdays, 9-10am**
- Labs on **Fridays, 9-11am**
  - Several labs are run at the same time (see next slide)
    - The situation with covid is fluid, therefore please pay attention to any changes and notices around physical labs
  - Labs provided via GIThub (we will learn together how to use it)
  - Lab tests submissions via brightspace

Sense of community: If you like, please update your bio with a short profile and a picture. Also, please configure your notification settings and install the VLE on your mobile.

Brightspace

- “Welcome” unit contains all basic info around the course
- All virtual classroom sessions are available in the “Virtual Classroom” unit for fast access
- All labs are available in the “The Labs” unit for fast access
- Every week has its own unit in the brightspace content area
    - Each week represents one topic
- A new week is open to you if you
    - Have completed the revision (checklist or self-test), **and**
    - And it’s the week you are in. Modules open at 8am on the day of the Monday lecture
- Check for announcements



TU856/858 - Year 2: Object Oriented Programming (CMPU2016) 2020-21 with Bianca and Bryan

15 people visited this course today.

No quizzes have been submitted today.

First Lecture

Posted 17 September 2020 1:36 PM

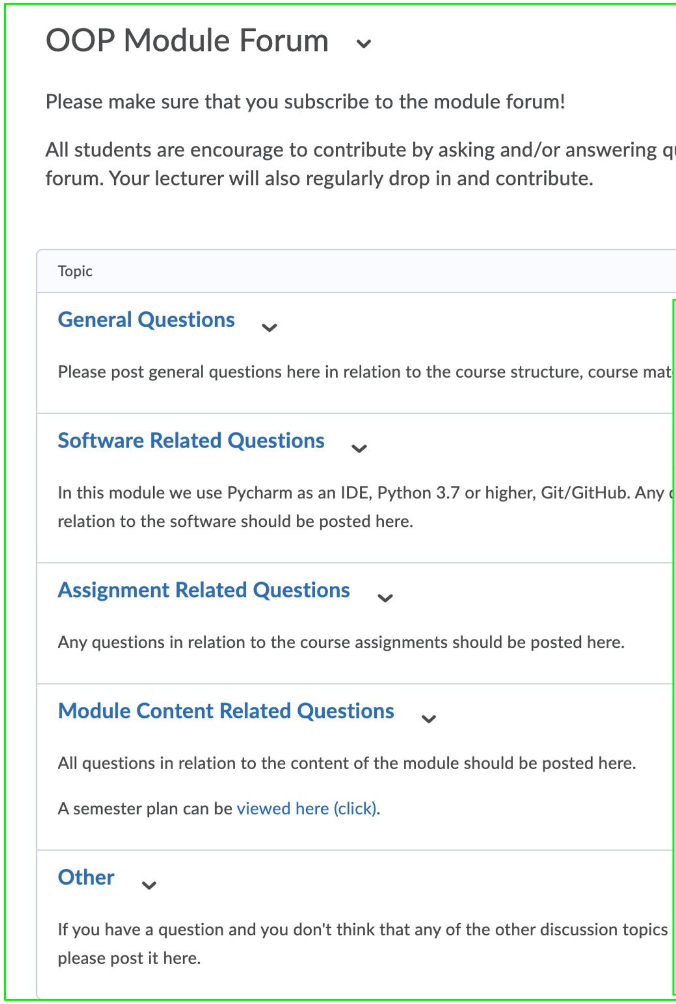
The first lecture in OOP with Python takes place on Monday, 21 September 2020 at 9am.

Communication Statement

Please subscribe to the discussion forum in brightspace!

- All information will be distributed via brightspace. At the beginning of the semester email lists will also be used to facilitate those students who don't have brightspace access yet.
- The announcements area will be used to announce any changes to the class, the schedule, etc.
- All virtual sessions are in the calendar.
- The calendar has the dates of lab test assignments.
- This course uses the discussion forum to let students discuss topics relating to the course. You can access the discussion forum via the task list at the top "Module Tools" > "Discussions". Your lecturer will regularly check the discussion forum and contribute. **Subscribe to the forum!**
- If you need to contact me, please use email: bianca.phelan@tudublin.ie
- I aim to respond to every email within 48 hours.

Subscribe to the forum



OOP Module Forum ⌄

Please make sure that you subscribe to the module forum!

All students are encourage to contribute by asking and/or answering questions in t
forum. Your lecturer will also regularly drop in and contribute.

Topic

General Questions ⌄

Please post general questions here in relation to the course structure, course mat

Software Related Questions ⌄

In this module we use Pycharm as an IDE, Python 3.7 or higher, Git/GitHub. Any
relation to the software should be posted here.

Assignment Related Questions ⌄

Any questions in relation to the course assignments should be posted here.

Module Content Related Questions ⌄

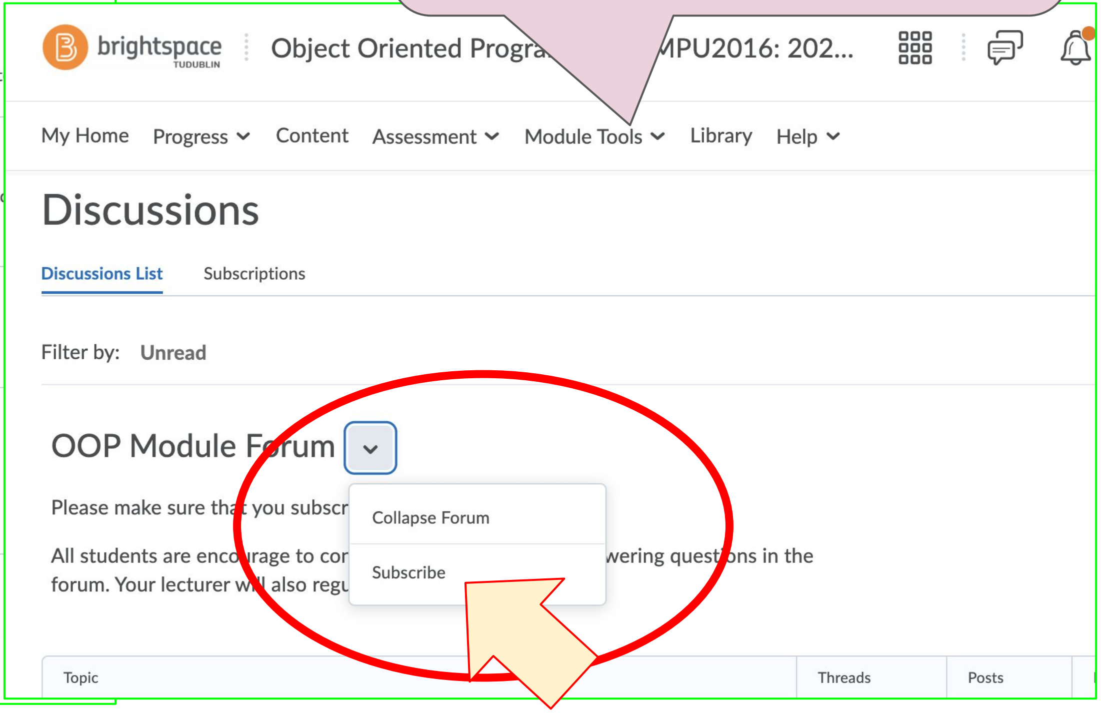
All questions in relation to the content of the module should be posted here.

A semester plan can be viewed here (click).

Other ⌄

If you have a question and you don't think that any of the other discussion topics
please post it here.

The discussion forum is accessible via **Module Tools > Discussions**



Discussions

Discussions List | Subscriptions

Filter by: Unread

OOP Module Forum ⌄

- Collapse Forum
- Subscribe

Please make sure tha you subscr

All students are enco rage to cor wering quest ons in the
forum. Your lecturer w ll also regu

Topic	Threads	Posts

Netiquette

1. **Ask questions**: You are encouraged to ask questions during all online sessions. Please start by writing your question in the chat or raise your hand to indicate that you have a question. Your lecturer will pause regularly to check for questions. You may be asked to speak as well.
2. **No yelling**: You are encouraged to ask questions during all online sessions using the chat window. Most readers tend to perceive it as shouting to use ALL CAPS and will have a hard time taking what is said seriously.
3. **Sarcasm can and will backfire**: Sarcasm has been the source of plenty of misguided arguments online, as it can be incredibly difficult to understand the commenter’s intent. It is best to avoid sarcasm altogether in an online classroom. Instead, lean toward being polite and direct in the way you communicate to avoid these issues.

Netiquette cont’d

4. **Don't change the topic**: Please keep all classroom chat to the topic discussed in the course module. The chat facility is indented to be a useful resource, not a distraction.

5. **RTFM**: Are you sure the information isn't already provided? Repeat questions on items such as classroom structure, assignment deadlines, etc. can become very tedious for all students to hear discussed repeatedly. Please make sure that the answer to your question hasn't been provided already in the documentation on brightspace.

6. **Read first**: Similar to the previous point, but relates to the chat window. Take some time to read through each of the previous discussion post responses before writing your own response. If the original post asked a specific question, there’s a good chance someone has already answered it. Submitting an answer that is eerily similar to a classmate’s indicates to the instructor that you haven’t paid attention to the conversation thus far. Remember, discussions can move fairly quickly so it’s important to absorb all of the information before crafting your reply. Building upon a classmate’s thought or attempting to add something new to the conversation will show your instructor you’ve been paying attention.

Netiquette cont’d

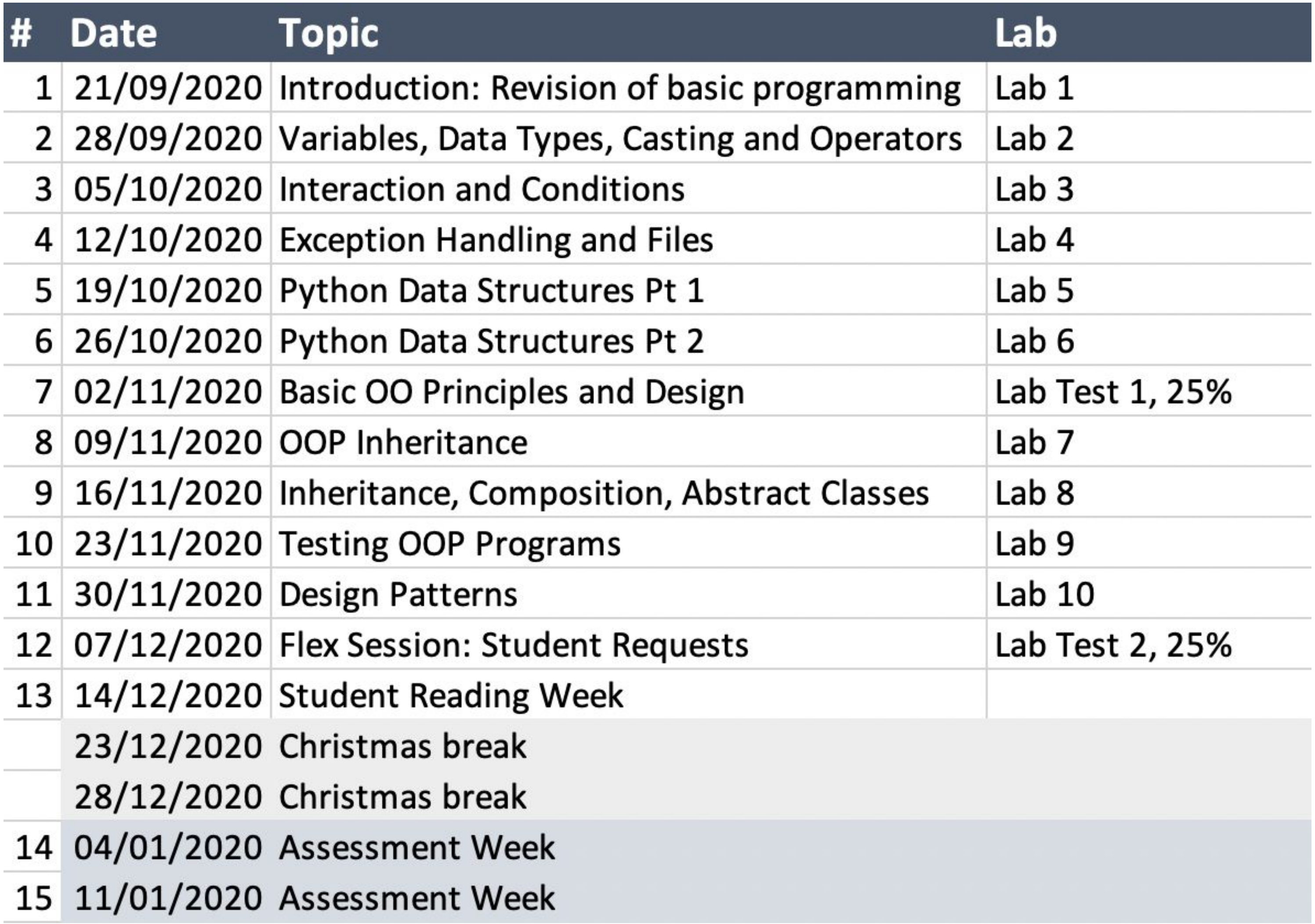
7. **Be kind and professional**: Online communication comes with a level of anonymity that doesn’t exist when you’re talking to someone face-to-face. Sometimes this leads people to behave rudely when they disagree with one another. Online students probably don’t have the complete anonymity that comes with using a screen name, but you could still fall prey to treating someone poorly because of the distance between screens. Make a point to be kind and respectful in your comments—even if you disagree with someone.

Labs

Depending on the situation with covid, this may change and we may be able to offer some labs physically in Kevin Street and Augnier Street. Changes will be announced on brightspace.

#	Room/Online	Tutor
1	Online	Bianca Schoen-Phelan
2	Online	Bryan Duggan
3	Online	Lucas Rizzo
4	Online	Sagar Saxena
5	Online	Sean O'Leary
6	Online	Giancarlo Salto

Semester Overview



#	Date	Topic	Lab
1	21/09/2020	Introduction: Revision of basic programming	Lab 1
2	28/09/2020	Variables, Data Types, Casting and Operators	Lab 2
3	05/10/2020	Interaction and Conditions	Lab 3
4	12/10/2020	Exception Handling and Files	Lab 4
5	19/10/2020	Python Data Structures Pt 1	Lab 5
6	26/10/2020	Python Data Structures Pt 2	Lab 6
7	02/11/2020	Basic OO Principles and Design	Lab Test 1, 25%
8	09/11/2020	OOP Inheritance	Lab 7
9	16/11/2020	Inheritance, Composition, Abstract Classes	Lab 8
10	23/11/2020	Testing OOP Programs	Lab 9
11	30/11/2020	Design Patterns	Lab 10
12	07/12/2020	Flex Session: Student Requests	Lab Test 2, 25%
13	14/12/2020	Student Reading Week	
	23/12/2020	Christmas break	
	28/12/2020	Christmas break	
14	04/01/2020	Assessment Week	
15	11/01/2020	Assessment Week	

Module Assessment

- Exam 50% : Continuous Assessment 50%
  - 50% semester 1
  - 50% semester 2
- Continuous assessment s1:
  - 2 lab tests, both worth 25%
  - Week 7 and week 12

Check semester week overview spreadsheet on brightspace (or previous slide).

Semester 1 is OOP via Python with me, Semester 2 is OOP via Java with Bryan Duggan.
We have **one joint** exam paper at the end of the year.

Module Descriptor

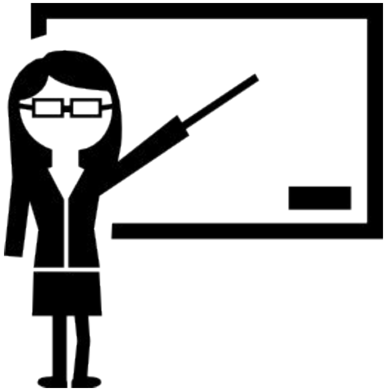
The aim of this module is that the student become familiar with Object Oriented Programming **concepts** and **implement** these concepts in **elementary** object oriented programs.

This is just a small citation from the module descriptor.

About Your Lecturer

Bianca Schoen-Phelan, PhD

- bianca.phelan@tudublin.ie
- http://biancaphelan.ie
- @BSPhelan
- Brief CV:
  - Since 2013: Lecturer at TU Dublin
  - 2010-2013: Software Engineer, SUN Oracle
  - 2010: Postdoc, NCG, NUI Maynooth
  - 2008: Postdoc, UCD
  - 2007: IT Consultant, Hewlett-Packard



Technology Used

- git/GIThub
- Python 3
- PyCharm

References

A concise introduction to programming in Python, Johnson, Mark, 2nd edition, CRC press, 2018.

Python in a nutshell, Martelli, Alex, 3rd edition, O’Reilly, 2017.

Test-driven development with python, Percival, Harry, 2nd edition, O’Reilly, 2017

Background OOP

General Background

- Programming was initially very **machine focused**
- We’ve quickly learnt that this is an **error prone** and difficult to grasp concept
- Object-oriented programming is **human focused** and aims to model the way we see, understand and interact with the **real world**

Machine Languages

- Lots of 1s and 0s
- Error prone because the placement of each 1 and 0 matters!
- Example code:

```
1110010110110      11101010100010
1011101011010100   100000001010111
```

Assembly Languages

- A set up from 0s and 1s
- A set of elementary commands
- The assembly code needs to be translated to machine language so that the computer can process it
- Example code:

```
ADD 1111001, 1110001
```

High-Level Languages

- Huge difference
- Constructed to help the human
- Easier to read, write and maintain
  - Syntax is similar to English
- Two types (historically speaking):
  - Procedural
  - Object-oriented

Procedural Languages

- Developed first
- Sequential sets of linear instructions
- Focus on structure
- Examples: C, Fortran, Lisp, Perl, HTML

Object-Oriented Languages

- Focus on modelling the real world
- Use of classes to model real world behaviours and interactions
- There are pure OO Languages and Hybrids
- Examples: Java, C++, C#, Python

OOP Terminology

Term	Description
Object	A person, place or thing (a noun)
Method	An action performed by an object, (a verb)
Property or Attribute	Characteristics of an object
Class	A template for creating objects

The real world can be accurately described by a collection of objects that interact.

Difference Class vs Object

- A class is a template or blueprint for creating an object
  - An object is an instance of a class
- Class is composed of:
  - Name
  - Attributes/properties
  - Methods

Object is a class in runtime.

Objects interact through messages to solve a problem.

- Homo sapiens is a class, Sean and Mary are objects
- Animal is a class, ‘Flash’ the terrier is an object
- Vehicle is a class, my Honda is an object
- Galaxy is a class, Milky Way is an object

Background Python

General History

- Created in the 1980s by Guido van Rossum
- Emphasis is code readability and simplicity
  - Allows for rapid code development
- Code is written in a language similar to English, which needs translating so that the computer understands:
  - Download python interpreter
  - There is a number of pyinstallers that package the programs for the various platforms

Why Python?

- Easy to learn
  - Less lines of code than many other languages
  - Reduced amount of errors
  - Less development time
- Strong library support
- Varied application areas
- Cross platform

Summary

- ★ **Course Overview**
- ★ **Assessment Structure**
- ★ **Weekly breakdown**
- ★ **Background Python**

References

1. Moutaz Haddara, Introduction to Object-oriented programming, slideshare, 2014.
2. Jamie Chan, Learn Python in one day, 2014.