cct | College Dublin
Computing • IT • Business

PROGRAMMING: OBJECT-ORIENTED APPROACH 

INTRODUCTION

- Press `Space` to navigate through the slides
- Use `Shift+Space` to go back
- Save as **PDF**:
  - Open **Chrome** then **click here**
  - Press **Ctrl+P**/**Cmd+P** to print
    - Destination: **Save as PDF**
    - Layout: **Landscape**
  - Press **Save** button

IN-CLASS TECHNOLOGIES 

- For communicating with me:
  - **Slack** – http://cct-dip-ai.slack.com
- For in-class demonstration:
  - **Slides** – https://mikhail-cct.github.io/ooapp/wk1
  - Change `wk1` to **any required week number** at the end of the **URL**
- Online IDE:
  - **GitPod.io** – https://gitpod.io
- Code hosting:
  - **GitHub** – https://github.com
- For all other needs:
  - **Moodle** – http://moodle.cct.ie

MODULE LEARNING OBJECTIVES

- Understand and employ fundamental concepts and principles of programming such as variables, Boolean expressions, control flow structures, methods, arrays, etc.
- Demonstrate a structured approach to algorithmic design and problem solving and exhibit professional development best practices in designing and developing robust, maintainable software
- Illustrate and relate object-oriented concepts (encapsulation, inheritance, polymorphism) and employ them to solve practical, real-world problems
- Differentiate, select and utilise suitable application programming interfaces in the construction of software
- Discriminate between elements of object-oriented programming (abstract and nested classes, interfaces, access modifiers, etc.) and employ them appropriately in programme construction

CODE STORAGE, EDITOR/IDE SETUP

- **GitHub**
  - Please register and let's create a first empty repository
- **GitPod**
  - You can start any GitHub repository by appending `https://gitpod.io#` to the repo URL

4

PYTHON FILES AND REPL

There are two different modes to run Python:

- **Files**
  - Python files use the .py extension on the file.
  - Python files are read top down and execute **line-by-line**
  - Unlike *compiled languages*, Python (since it is *interpreted*) does not check **logical errors** before running your code.
  - This means any bugs that are written will only be caught by getting to that point in the running file.
- **REPL (Read Evaluate Print Loop)**
  - Python has an advantage over other languages in that you can run Python code without having it in a file.
  - You can use a REPL environment to run pieces of Python code on the fly, with immediate feedback and without having to create & run a file.

REPL (READ EVALUATE PRINT LOOP)

- On Windows, Mac and Linux you can run `Python` and should get a prompt like this:

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- From here you can start typing Python code in, and it will run line by line as you type it.

PYTHON BASIC SYNTAX: COMMENTS

- In Python you can add comments (text that doesn't do anything)
- This is incredibly useful for leaving yourself and others notes about how code works, what code does, or to 'comment out' code that you just don't want to run
- Throughout the course most of the Python files will contain comments that will help you to understand what is happening in the code.
- Also, I will leave sections of challenges and exercises 'commented out' to allow you to work on them in order.

> *"Commenting your code is like cleaning your bathroom – you never want to do it, but it really does create a more pleasant experience for you and your guests" - Ryan Campbell*

COMMENTS

There are two ways of doing comments:

1. Single line comments; comments that only span a single line are denoted with a # in front of them:

```
# This is a single line comment
```

2. Multiline comments; comments that span multiple lines are denoted with three sets of double quotes:

```
"""
This
comment
spans
many
lines
"""
```

- As you are going through it is always a good idea to put comments in your code so that other people (and you in 3 months) will know what's happening.
- Believe me this **will** make a difference

8

COMMENTS

True programmers don't comment their code... If it was hard to write then it should be hard to read – Anonymous

When I wrote this code only God and I understood what it does... Now, only God knows. – People who don't comment their code

FUNCTIONS

- Functions in Python are commands you can use to do specific actions.
- Functions can also be given data (called arguments), and return data.
- The basic syntax looks like this

```
function-name(arguments)
```

FUNCTIONS

- You can tell that something is a function if it has parenthesis "()" after the function name.
- For example, the **`print()`** function in Python takes some text (a *string* [i'll explain what that is in the next lecture] as an argument) and prints it to the terminal.

```
print("Hello World!")
```

11

RUNNING PYTHON CODE

To run your code (after you've written it) use: `python (filename).py`

EXERCISE TIME

Check out the exercises.py for some simple exercises to try out.