University of Essex

Online

Launching into Cyber Security

Week 8 Seminar

University of Essex

Online

Announcement

- Collaborative discussion 2

- Individual essay – formative feedback

University of Essex

Online

Discussion outline

- Intro to Python
- Intro to MySQL
- Integration of Python and MySQL
- Application to PACs

University of Essex

Online

Discussion outline

- **Intro to Python**
- Intro to MySQL
- Integration of Python and MySQL
- Application to PACs

University of Essex

Online

Intro to Python

- Python began in 1989
- High level - scripting language
- Supports object oriented programming
- Scalable – pluggable & modular architecture
- Extensible – core security modules can be written in another language
- Portable across multiple architectures and operating systems – ANSI C compiler
- Easy to learn - few keywords
- Easy to read – understandable to a non programmer
- Easy to maintain

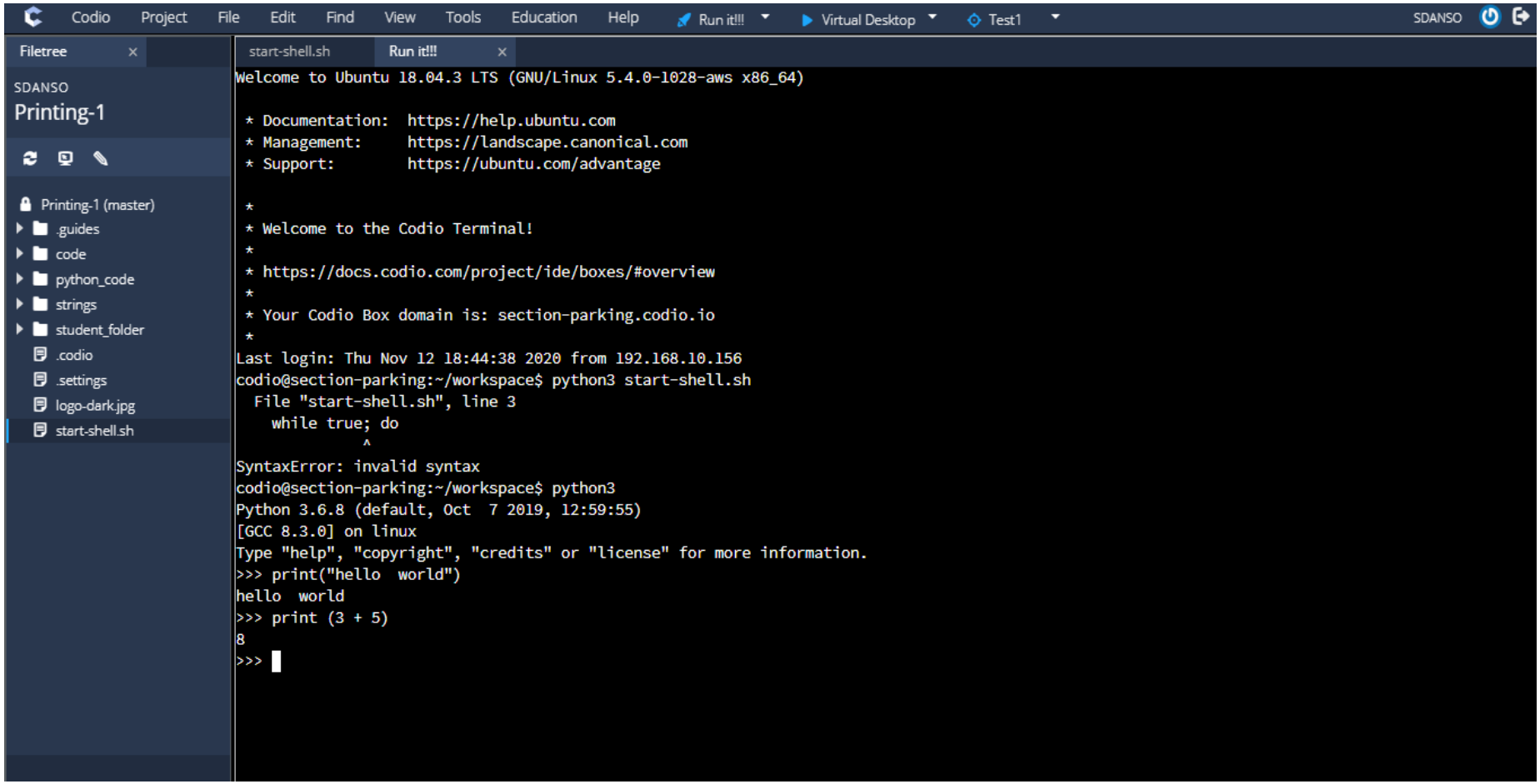
Intro to Python

- Installation options
  - Unix-based systems
  - Windows/DOS
  - Build it yourself
- Running
  - Interactive
  - Running as a script
  - From GUI via IDE which has additional features

University of Essex

Online

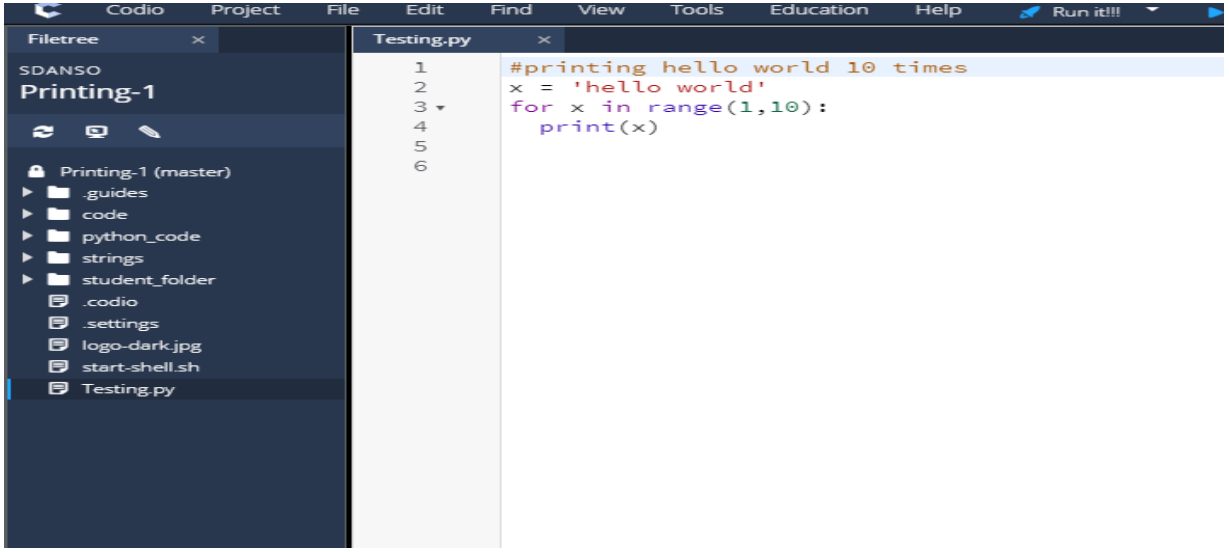
Intro to Python: interactive



University of Essex

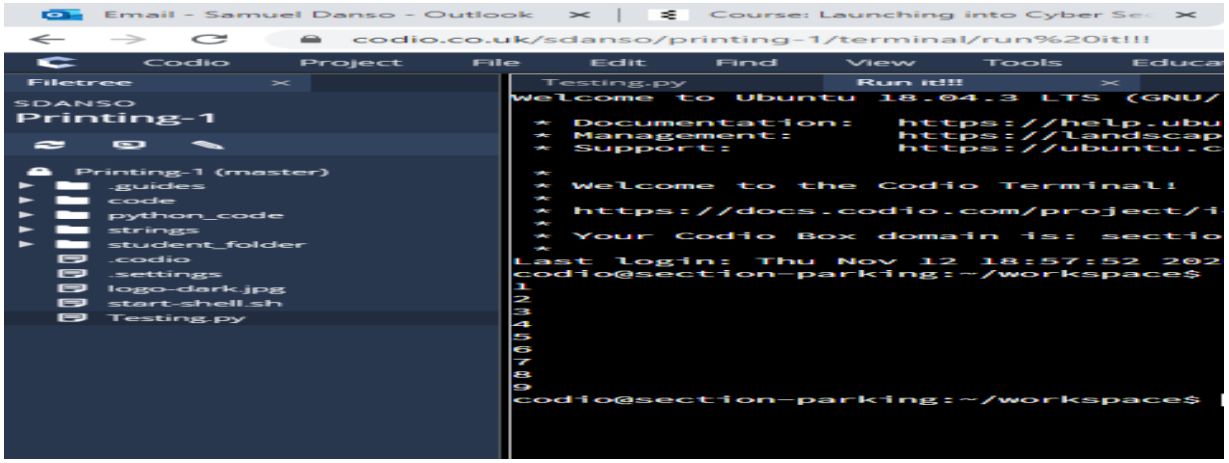
Online

Intro to Python: script



Script:

- Filename ..e.g testing.py
- Code
- Comments



Output

University of Essex

Online

Intro to Python: IDE

Integrated Development Environment

Open source

- IDLE
- PythonWin
- Ipython
- IDE studio
- Eclipse
- Text editors
- Codio

Commercial

- WingIDE Python IDE by WingWare
- Komodo IDE by ActiveState

Intro to Python: getting started

- Output and print statement

```
>>myString = 'hello world'
>>print (mystring)
  hello world
```

- Input and the raw_input() built-in function

```
>>user = raw_input ('Enter login name: ')
Enter login name: root
>> print ('Your login is: ', user)
Your login is: root
```

University of Essex

Online

Intro to Python: getting started

- **Comments**

  >>\# my first comment

  >>print ('hello world') # my other commnet

  hello world

- **Operators**

  ❑ + - * / // % **

  >> print (-2 * 4 + 3 ** 2)

  1

  ❑ < <= > >= == != <>

  >> 2 < 4

  True

  ❑ and or not

  >> 2 < 4 and 2 == 4

  False

Intro to Python: getting started

- **Variables and assignments**
  - case sensitive;
  - no pre-declaration;
  - type (and value) are assigned during initialisation
  - equal signs are used for assignments

```
>> counter = 0
>> miles = 1000.0
>> name = 'John'
>> counter = counter + 1
>> kilometers  = 1.609 * miles
>> print ('%f miles is the same as %f km' % (miles, kilometers) )
```

1000.00000 miles is the same as 1609.000000 km

Intro to Python: getting started

- **Numbers**
  - int (signed integers) - e.g 0101 84 -353 -0x80 017 -0X92
  - int bool (Boolean value) - e.g True, False
  - float (floating point real numbers) 3.14159 4.2E-10 -90. 6.022e23
  - Complex - e.g square root of -1 (6.23+1.5j)
  - Decimal - need importing decimal module

```
>> print(decimal.Decimal('1.1')
1.1
```

Intro to Python: getting started

- **Strings**
  - Continues set of characters in between double or single quotation marks
  - Subsets of strings can extracted using index ([ ]) or slice ([:])

```
>> pytstr =  'Python'
>> iscool = 'is cool!'
>>pystr[0]
'P'
>>pystr[2:5]
'tho'
```

Intro to Python: getting started

- **List and Tuples**
  - Generic arrays used to hold a Python objects
  - Items are ordered and are accessible via index offsets
  - Lists are enclosed in brackets ([ ]) and their elements can be changed
  - Tuples are enclosed in parenthesis (( )) and cannot be updated - they are read-only
  - Subsets of can be taken with slice operator ([ ]) and ([:])

```
>>aList = [ 1, 2, 3, 4]
>> aList
[ 1, 2, 3, 4]
>> aList[0]
[1]
>>aList[2:]
[ 3, 4]
```

Intro to Python: getting started

```
>>aTuple = ('robots', 77, 99,  'try')
>>aTuple
('robots',  77,  99 , 'try')
>>aTuple[:3]
('robots',  77,  99 )

>>aList[1] = 5
[ 1, 5, 3, 4]

>>aTuple[1] = 5  # ???
```

University of Essex

Online

Intro to Python: getting started

- **Dictionaries**
  - ‘dict’ for short
  - Made up of key-value pairs
  - Keys are numbers of strings
  - Enclosed by curly brackets ({ })

```
>> aDict = {‘username’: ‘root’ }
>>aDict[‘port’] = 80
>> aDict
{‘username’: ‘root’,  ‘port’: 80 }
>>aDict.keys()
[‘username’, ‘port’ ]
>> aDict[‘username’]
‘root’
```

Intro to Python: getting started

- **Looping over dictionaries**

```
>> for key in aDict:
        print (key, aDict[key] )
```

```
username  root
port 80
```

Intro to Python: getting started

Conditional statement:

- If statement
  - ❑ if and else;
  - ❑ if, elif and else

- While loop

```
>>counter = 0
>> while counter< 3:
        print (counter)
       counter += 1
```

Intro to Python: getting started

- **Opening files**

```
>>  filename = raw_input('Enter file name')
>>fobj = open (filename, 'r')
>>data = jobj.readlines()
>>fobj.close()
>> for eachLine in  data:
        print eachLine,
```

University of Essex

Online

Intro to Python: getting started

Functions

- **Inbuilt functions:**
  - file ()
  - open()

- **Self-declared functions:**

```
>> def addMe2Me(x):
        'apply + operation to argument '
        return (x + x)
```

- **Calling functions:**

```
>> addMe2Me(7)
14
```

Intro to Python: getting started

Classes

- A core part of OOP and serves as the container for related data and logic
- They provide a mechanism for creating objects called instances
- Have attributes
- Have optional documentation string
- May have methods (functions) declarations

Intro to Python: getting started

Class example

```
class FooClass(object):
        """my very first class: FooClass"""
         version = 0.1
def __init__(self, nm='John Doe'):
         ""constructor""
         self.name=nm
         print ('Create a class instance for', nm)
def showname (self):
        print( 'Your name is' , self.name)
def addMe2Me (self, x):
       return  x + x
```

Intro to Python: getting started

Using the Class

```
>> fool = FooClass ()


>> fool.showname ()
Your name is John Doe


>>print(fool.addMe2Me(6) )
12
```

University of Essex

Online

Intro to Python: getting started

Modules

- A logical way to physically organise and distinguish related Python code into individual files.
- May contain executable code, functions, classes.

```
>>import sys

>>sys.stdout.write('Hello World!\n')

Hello World!

>>sys.platform

'win32'
```

University of Essex

Online

Intro to Python: getting started

List of Python keywords				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

source: https://www.programiz.com/python-programming/keyword-list

University of Essex

Online

Discussion outline

- Intro to Python
- **Intro to MySQL**
- Integration of Python and MySQL
- Application to PACs

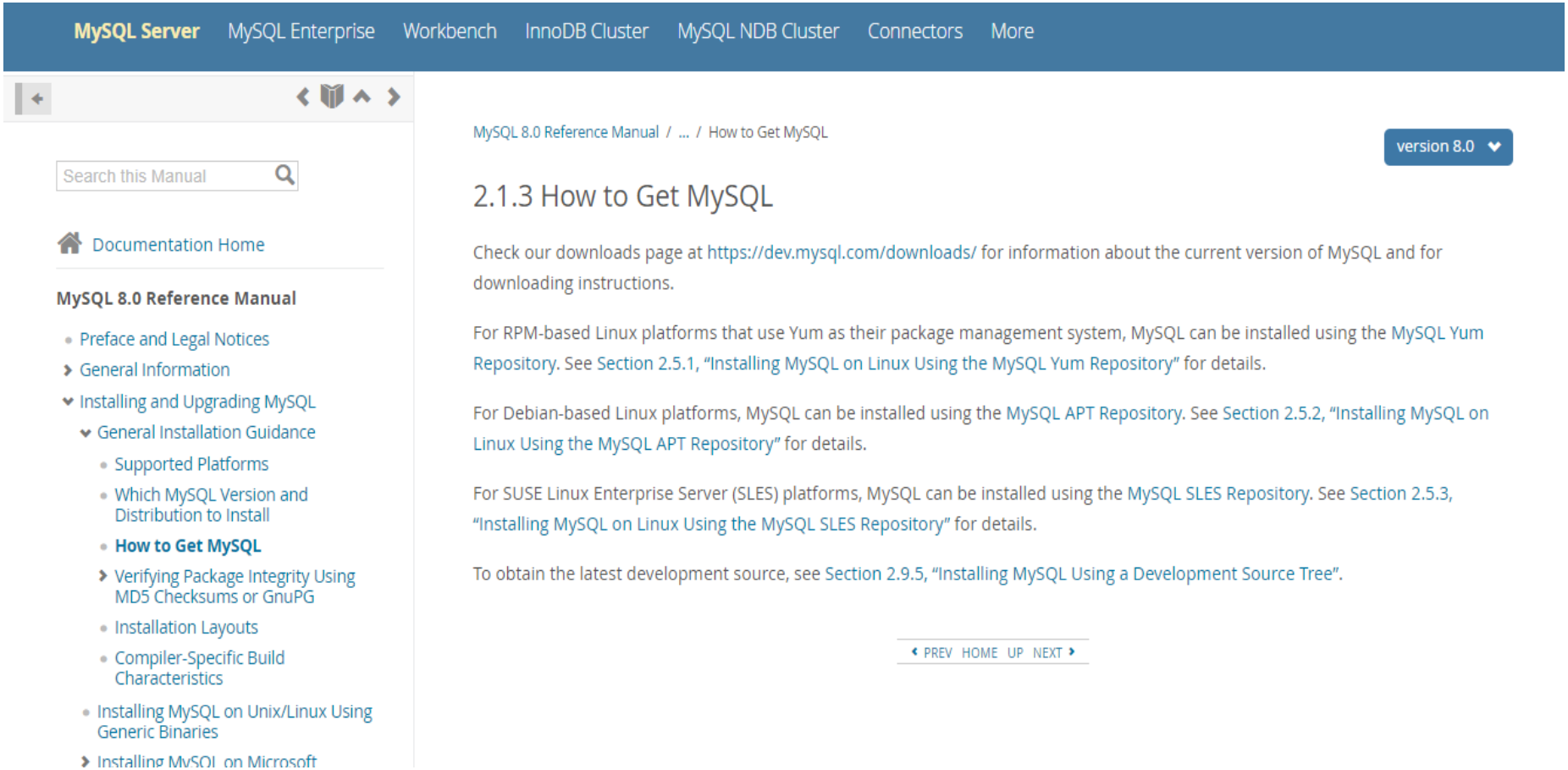
Intro to MySQL

- A database requires a database management system
- MySQL is a Structured Query Language (SQL) based relational database management system
- MySQL is compatible with standard SQL
- Mostly used by PHP and Perl, and now Python

University of Essex

Online

Intro to MySQL



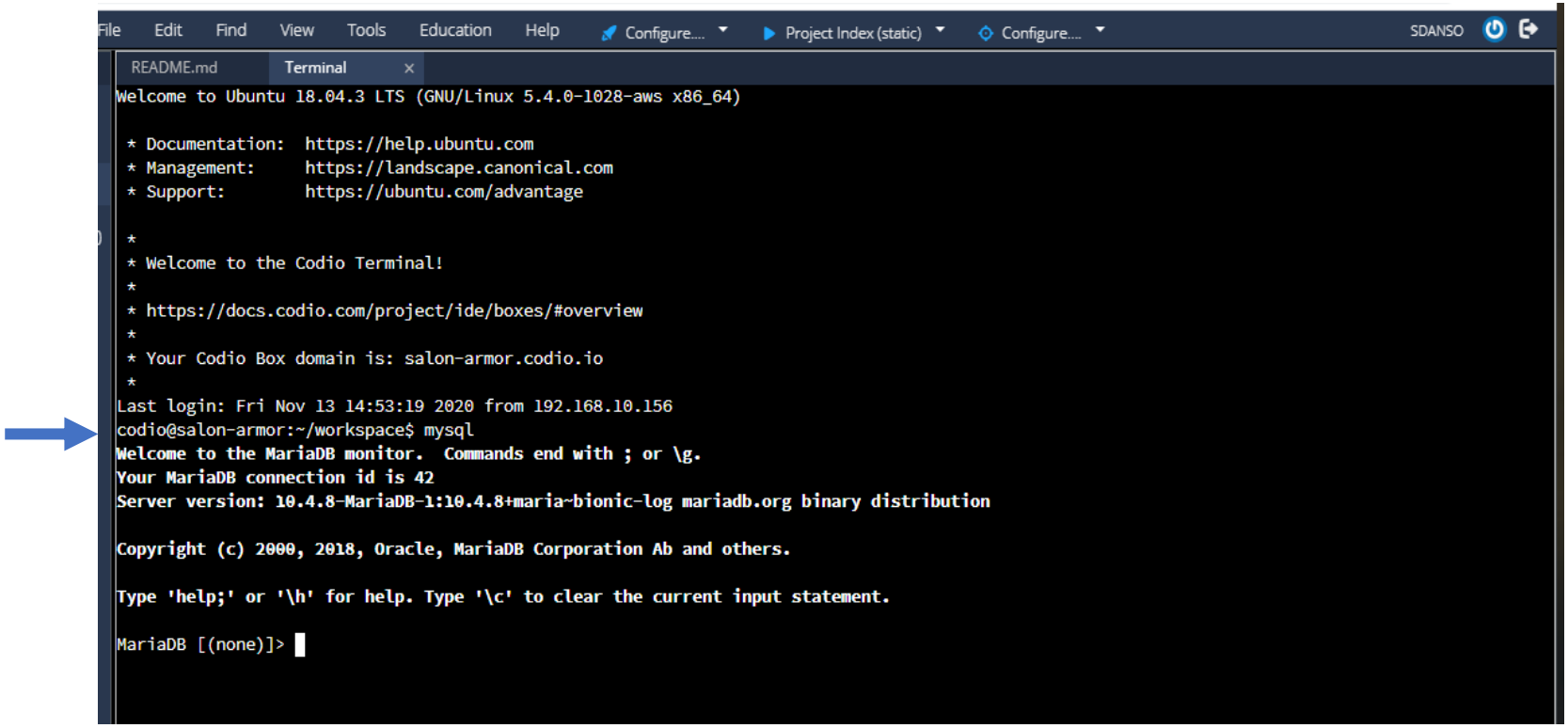
University of Essex

Online

Intro to MySQL: accessing MySQL

>mysql –u [username] –p

>Enter password:[password]



University of Essex

Online

Intro to MySQL: prompts and meaning

prompt	meaning
mysql>	Ready for new command.
->	Waiting for next line of multiple-line command.
'>	Waiting for next line, waiting for completion of a string that began with a single quote ("'").
">	Waiting for next line, waiting for completion of a string that began with a double quote ("").
`>	Waiting for next line, waiting for completion of an identifier that began with a backtick ("`").
/*>	Waiting for next line, waiting for completion of a comment that began with /*.

Taken from: kent.edu

Intro to MySQL: command

- Prompt (mysql>) sends it to the server for execution
- mysql> displays the results
- Prints another mysql>
- A command could span multiple lines
- A command show have an SQL statement followed by a semicolon
- Help---------> \h
- Quit--------->/exit or \q
- Cancel the command ---------> \c
- Change database ---------> use

University of Essex

Online

Intro to MySQL: info about databases and tables

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| directory          |
| information_schema |
| mysql              |
| people             |
| performance_schema |
+--------------------+
5 rows in set (0.000 sec)

MariaDB [(none)]> use people;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [people]> show tables;
+----------------+
| Tables_in_people |
+----------------+
| basic_info     |
| names          |
+----------------+
2 rows in set (0.000 sec)

MariaDB [people]> describe names;
+---------+-----------------------+------+-----+---------+----------------+
| Field   | Type                  | Null | Key | Default | Extra          |
+---------+-----------------------+------+-----+---------+----------------+
| id      | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| name    | varchar(255)          | YES  |     | NULL    |                |
| surname | varchar(255)          | YES  |     | NULL    |                |
+---------+-----------------------+------+-----+---------+----------------+
3 rows in set (0.001 sec)

MariaDB [people]>
```

University of Essex

Online

Intro to MySQL: query

```
MariaDB [people]> select * from names;
+----+----------+----------+
| id | name     | surname  |
+----+----------+----------+
|  1 | Pamela   | Lyons    |
|  2 | Daria    | Kelley   |
|  3 | Morgan   | Aguirre  |
|  4 | Angelica | Mccarthy |
|  5 | Xena     | French   |
|  6 | Colby    | Albert   |
|  7 | Xenos    | Duke     |
|  8 | Clark    | Shields  |
|  9 | Tasha    | Steele   |
| 10 | Yeo      | Britt    |
+----+----------+----------+
10 rows in set (0.000 sec)

MariaDB [people]> select * from names where id = 5;
+----+------+---------+
| id | name | surname |
+----+------+---------+
|  5 | Xena | French  |
+----+------+---------+
1 row in set (0.000 sec)

MariaDB [people]>
```

University of Essex

Online

Intro to MySQL: data from multiple tables using relational joins

README.md Terminal Terminal

```
+----+----------+----------+
10 rows in set (0.000 sec)

MariaDB [people]> select * from names where id = 5;
+----+------+---------+
| id | name | surname |
+----+------+---------+
|  5 | Xena | French  |
+----+------+---------+
1 row in set (0.000 sec)

MariaDB [people]> show tables;
+-------------------+
| Tables_in_people  |
+-------------------+
| basic_info        |
| names             |
+-------------------+
2 rows in set (0.000 sec)

MariaDB [people]> describe basic_info;
+----------+-----------------------+------+-----+---------+----------------+
| Field    | Type                  | Null | Key | Default | Extra          |
+----------+-----------------------+------+-----+---------+----------------+
| id       | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| email    | varchar(255)          | YES  |     | NULL    |                |
| birthday | varchar(255)          | YES  |     | NULL    |                |
| country  | varchar(100)          | YES  |     | NULL    |                |
| city     | varchar(255)          | YES  |     | NULL    |                |
+----------+-----------------------+------+-----+---------+----------------+
5 rows in set (0.001 sec)

MariaDB [people]> select * from basic_info a, names b where a.id=b.id;
+----+-----------------------------------------+------------+---------------------+---------------------+----+---------+-----------+
| id | email                                   | birthday   | country             | city                | id | name    | surname   |
+----+-----------------------------------------+------------+---------------------+---------------------+----+---------+-----------+
|  1 | erat.semper.rutrum@ipsumCurabitur.com   | 2014-12-15 | Sweden              | Bellevue            |  1 | Pamela  | Lyons     |
|  2 | Proin.sed.turpis@scelerisqueneque.ca    | 2015-08-06 | Pakistan            | Fontaine-l'Ev◆que   |  2 | Daria   | Kelley    |
|  3 | tincidunt.tempus.risus@acurnaUt.com     | 2016-03-05 | Comoros             | Enines              |  3 | Morgan  | Aguirre   |
|  4 | sed.pede@nisi.com                       | 2016-07-17 | Kiribati            | Frankenthal         |  4 | Angelica | Mccarthy |
|  5 | adipiscing@lorem.net                    | 2015-02-23 | Netherlands         | Springfield         |  5 | Xena    | French    |
|  6 | imperdiet.nec.leo@nec.ca                | 2016-07-22 | Hong Kong           | Richmond            |  6 | Colby   | Albert    |
|  7 | fermentum.vel@risusDonec.co.uk          | 2015-01-16 | Jersey              | Flin Flon           |  7 | Xenos   | Duke      |
|  8 | vel@natoquepenatibuset.edu              | 2015-09-27 | Hungary             | Vanier              |  8 | Clark   | Shields   |
|  9 | Suspendisse.eleifend.Cras@justonecante.net | 2014-12-18 | Antigua and Barbuda | Alken               |  9 | Tasha   | Steele    |
| 10 | non.cursus@duinec.edu                   | 2016-04-17 | Guatemala           | Ilbono              | 10 | Yeo     | Britt     |
+----+-----------------------------------------+------------+---------------------+---------------------+----+---------+-----------+
10 rows in set (0.000 sec)

MariaDB [people]>
```

University of Essex

Online

Intro to MySQL: database modifications

```
MariaDB [people]> alter table basic_info add employmentstatus varchar(255);
Query OK, 0 rows affected (0.092 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [people]> describe basic_info;
+------------------+-----------------------+------+-----+---------+----------------+
| Field            | Type                  | Null | Key | Default | Extra          |
+------------------+-----------------------+------+-----+---------+----------------+
| id               | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| email            | varchar(255)          | YES  |     | NULL    |                |
| birthday         | varchar(255)          | YES  |     | NULL    |                |
| country          | varchar(100)          | YES  |     | NULL    |                |
| city             | varchar(255)          | YES  |     | NULL    |                |
| employmentstatus | varchar(255)          | YES  |     | NULL    |                |
+------------------+-----------------------+------+-----+---------+----------------+
6 rows in set (0.001 sec)

MariaDB [people]> update basic_info set employmentstatus = 'Yes' where id=3;
Query OK, 1 row affected (0.021 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [people]> select * from basic_info;
+----+-----------------------------------------+------------+--------------------+--------------------+------------------+
| id | email                                   | birthday   | country            | city               | employmentstatus |
+----+-----------------------------------------+------------+--------------------+--------------------+------------------+
|  1 | erat.semper.rutrum@ipsumCurabitur.com   | 2014-12-15 | Sweden             | Bellevue           | NULL             |
|  2 | Proin.sed.turpis@scelerisqueneque.ca    | 2015-08-06 | Pakistan           | Fontaine-l'Ev◆que  | NULL             |
|  3 | tincidunt.tempus.risus@acurnaUt.com     | 2016-03-05 | Comoros            | Enines             | Yes              |
|  4 | sed.pede@nisi.com                       | 2016-07-17 | Kiribati           | Frankenthal        | NULL             |
|  5 | adipiscing@lorem.net                    | 2015-02-23 | Netherlands        | Springfield        | NULL             |
|  6 | imperdiet.nec.leo@nec.ca                | 2016-07-22 | Hong Kong          | Richmond           | NULL             |
|  7 | fermentum.vel@risusDonec.co.uk          | 2015-01-16 | Jersey             | Flin Flon          | NULL             |
|  8 | vel@natoquepenatibuset.edu              | 2015-09-27 | Hungary            | Vanier             | NULL             |
|  9 | Suspendisse.eleifend.Cras@justonecante.net | 2014-12-18 | Antigua and Barbuda | Alken              | NULL             |
| 10 | non.cursus@duinec.edu                   | 2016-04-17 | Guatemala          | Ilbono             | NULL             |
+----+-----------------------------------------+------------+--------------------+--------------------+------------------+
10 rows in set (0.000 sec)

MariaDB [people]>
```

University of Essex

Online

Database design: creating database and tables

```
CREATE DATABASE databasename;


CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

For more on SQL:

https://www.w3schools.com/sql/sql_create_db.asp

Database design considerations

- Content
  - Data to be stored and associated cost
- Access
  - Security to ensure who have access to what data
- Logical structure
  - Assembling the data to make sense to the user
- Physical organisation
  - The physical location of data storage

University of Essex

Online

Discussion outline

- Intro to Python
- Intro to MySQL
- **Integration of Python and MySQL**
- Application to PACs

University of Essex

Online

Database design: integration with Python

```
>>import mysql.connector

>>from mysql.connector import errorcode

>>conn = mysql.connector.connect(host="localhost", user="john", password='***')

>>curs = conn.cursor()

>>db = 'use people'

>>curs.execute(db)
```

Database design: integration with Python

```
def  addNewMember(id, email,  birthday, country, employmentstatus):
        id = id
        email = email
        birthday = birthday
        country = country
        status = employmentstatus
        to_db = [id, email, birthday, country, status]
        curs.execute("insert into basic_info (id, email, birthday, country, status)
values (%s,%s,%s,%s,%s); ",    to_db)
        conn.commit()
```

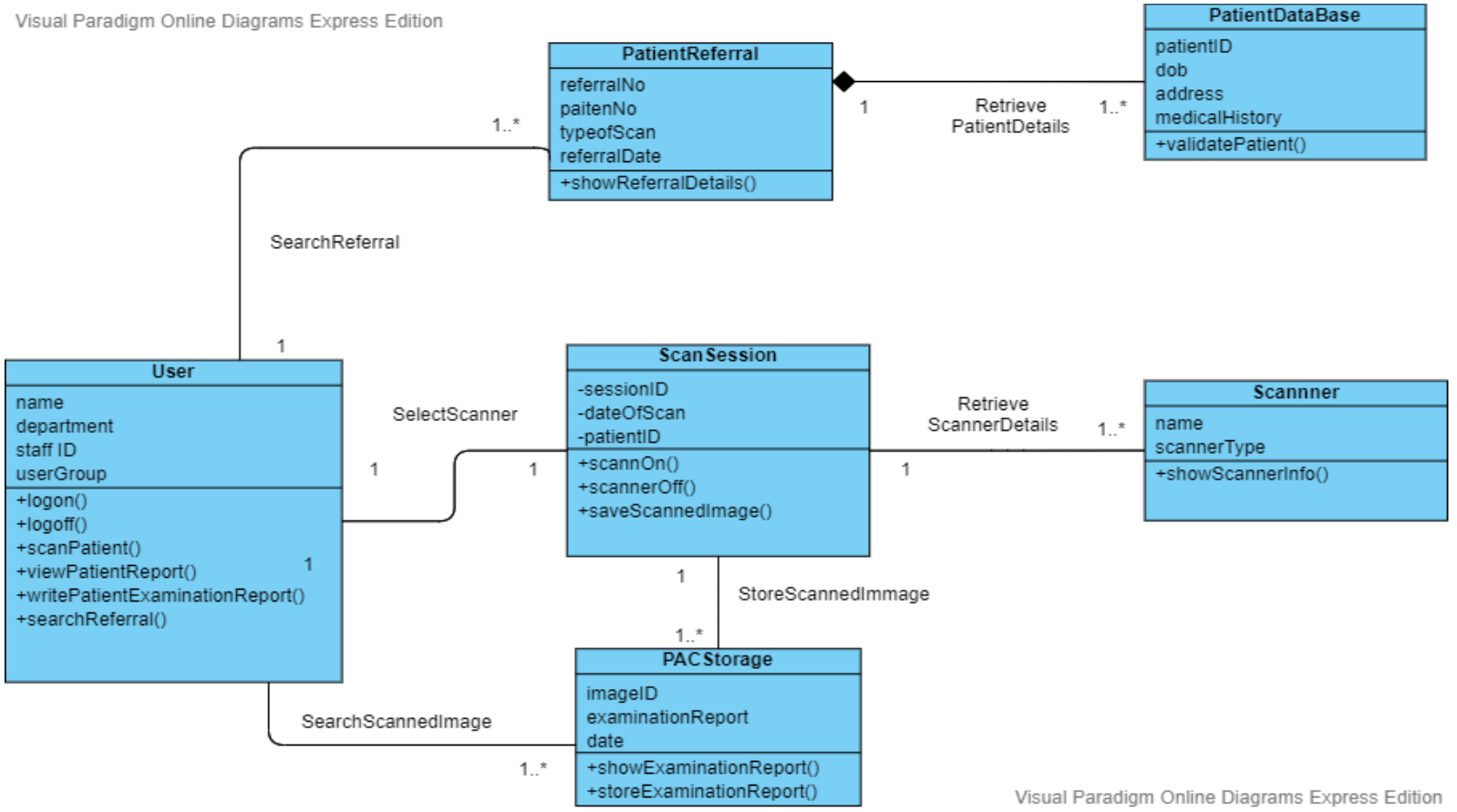
University of Essex

Online

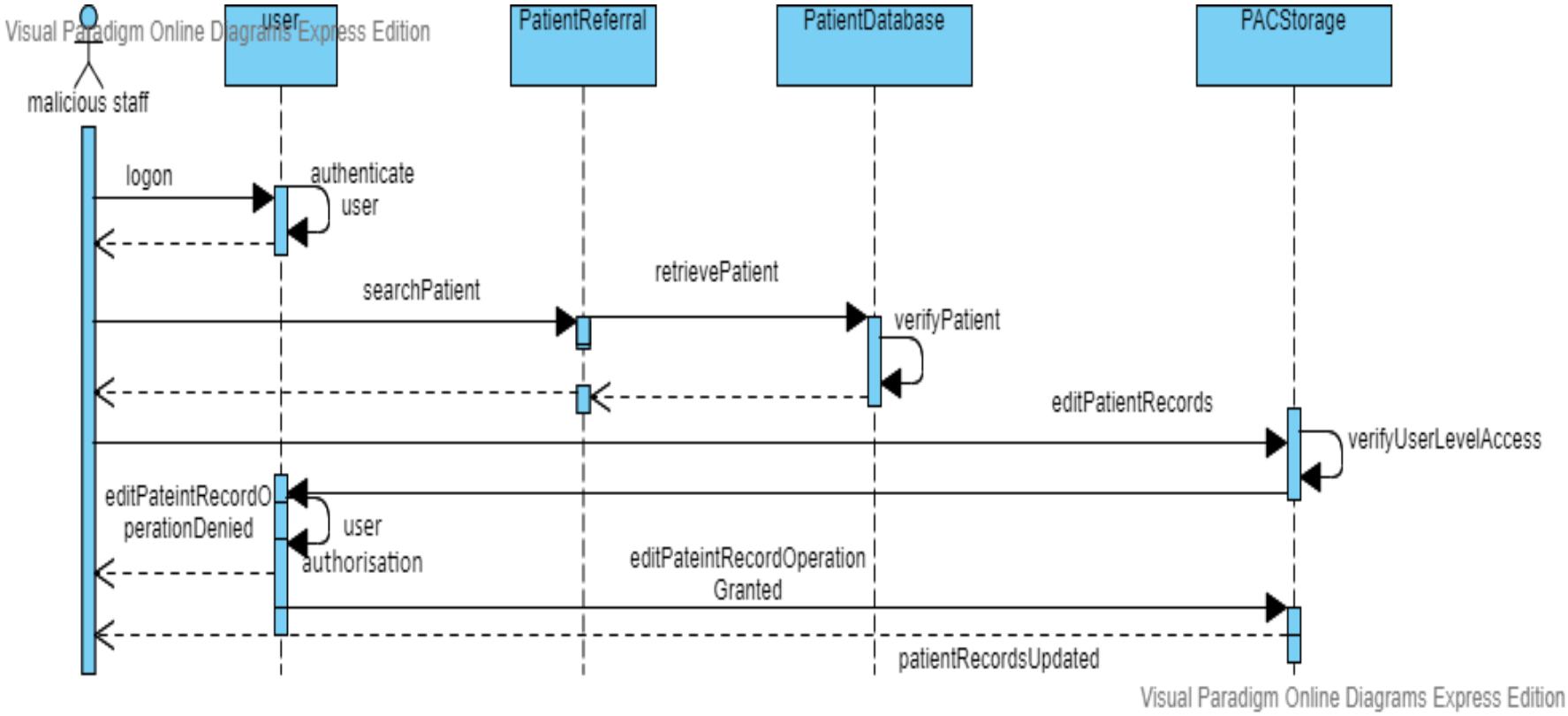
Discussion outline

- Intro to Python
- Intro to MySQL
- Integration of Python and MySQL
- **Application to PACs**

Application to PACs



Application to PACs



Exercise

Implement PACs solution using OOP with Python and MySQL