



University of Essex

Online

Launching into Cyber Security

Week 8 Seminar



Announcement

- Collaborative discussion 2
- Individual essay – formative feedback



Discussion outline

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



Discussion outline

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



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



Intro to Python: interactive

The screenshot shows the Codio IDE interface. The top menu bar includes Filetree, Project, File, Edit, Find, View, Tools, Education, Help, Run it!!!, Virtual Desktop, and Test1. The left sidebar shows the Filetree with a project named 'SDANSO' and a file named 'Printing-1'. The main terminal window displays the following output:

```
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.4.0-1028-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: section-parking.codio.io
*
Last login: Thu Nov 12 18:44:38 2020 from 192.168.10.156
codio@section-parking:~/workspace$ python3 start-shell.sh
File "start-shell.sh", line 3
    while true; do
        ^
SyntaxError: invalid syntax
codio@section-parking:~/workspace$ python3
Python 3.6.8 (default, Oct 7 2019, 12:59:55)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> print(3 + 5)
8
>>> 
```



Intro to Python: script

Script:

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

Output

The screenshot shows the Codio IDE interface. On the left is a Filetree panel showing a project named 'Printing-1' with various folders and files. The main editor window displays a Python script named 'Testing.py' with the following code:

```
1 #printing hello world 10 times
2 x = 'hello world'
3 for x in range(1,10):
4     print(x)
5
6
```

The screenshot shows the Codio IDE interface with the 'Run it!!!' button highlighted. The output terminal displays the following text:

```
Welcome to Ubuntu 18.04.3 LTS (GNU/
* Documentation:  https://help.ubu
* Management:    https://landscap
* Support:        https://ubuntu.c
*
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/1
* Your Codio Box domain is: sectio
Last login: Thu Nov 12 18:57:52 202
codio@section-parking:~/workspace$
codio@section-parking:~/workspace$
```




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



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 be 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 # ???
```




Intro to Python: getting started

• Dictionaries

- 'dict' for short
- Made up of key-value pairs
- Keys are numbers or 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 = fobj.readlines()
```

```
>> fobj.close()
```

```
>> for eachLine in data:  
    print eachLine,
```



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)
```



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



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'
```



Intro to Python: getting started

List of Python keywords

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

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



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



Intro to MySQL

MySQL Server MySQL Enterprise Workbench InnoDB Cluster MySQL NDB Cluster Connectors More



Search this Manual



Documentation Home

MySQL 8.0 Reference Manual

- Preface and Legal Notices
- ▶ General Information
- ▼ Installing and Upgrading MySQL
 - ▼ General Installation Guidance
 - Supported Platforms
 - Which MySQL Version and Distribution to Install
 - **How to Get MySQL**
 - ▶ Verifying Package Integrity Using MD5 Checksums or GnuPG
 - Installation Layouts
 - Compiler-Specific Build Characteristics
 - Installing MySQL on Unix/Linux Using Generic Binaries
 - ▶ Installing MySQL on Microsoft

MySQL 8.0 Reference Manual / ... / How to Get MySQL

version 8.0 ▼

2.1.3 How to Get MySQL

Check our downloads page at <https://dev.mysql.com/downloads/> for information about the current version of MySQL and for downloading instructions.

For RPM-based Linux platforms that use Yum as their package management system, MySQL can be installed using the MySQL Yum Repository. See Section 2.5.1, “Installing MySQL on Linux Using the MySQL Yum Repository” for details.

For Debian-based Linux platforms, MySQL can be installed using the MySQL APT Repository. See Section 2.5.2, “Installing MySQL on Linux Using the MySQL APT Repository” for details.

For SUSE Linux Enterprise Server (SLES) platforms, MySQL can be installed using the MySQL SLES Repository. See Section 2.5.3, “Installing MySQL on Linux Using the MySQL SLES Repository” for details.

To obtain the latest development source, see Section 2.9.5, “Installing MySQL Using a Development Source Tree”.


◀ PREV HOME UP NEXT ▶



Intro to MySQL: accessing MySQL

>mysql -u [username] -p

>Enter password:[password]



```
File Edit Find View Tools Education Help Configure.... Project Index (static) Configure.... SDANSO
README.md Terminal x
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.4.0-1028-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: salon-armor.codio.io
*
Last login: Fri Nov 13 14:53:19 2020 from 192.168.10.156
codio@salon-armor:~/workspace$ mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.4.8-MariaDB-1:10.4.8+maria-bionic-log mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```



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 /*.



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 should have an SQL statement followed by a semicolon
- Help-----> \h
- Quit-----> /exit or \q
- Cancel the command -----> \c
- Change database -----> use



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]> █
```



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]> 
```



Intro to MySQL: data from multiple tables using relational joins

```

+-----+-----+-----+
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@scelerisqueeneque.ca | 2015-08-06 | Pakistan               | Fontaine-L'Evêque  | 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]>

```



Intro to MySQL: database modifications

```

Error 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the
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@scelerisqueeneque.ca | 2015-08-06 | Pakistan | Fontaine-l'Évêque | NULL |
| 3 | tincidunt.tempus.risus@acurnaUt.com | 2016-03-05 | Comoros | Ennes | 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]> 

```



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



Discussion outline

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



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()
```



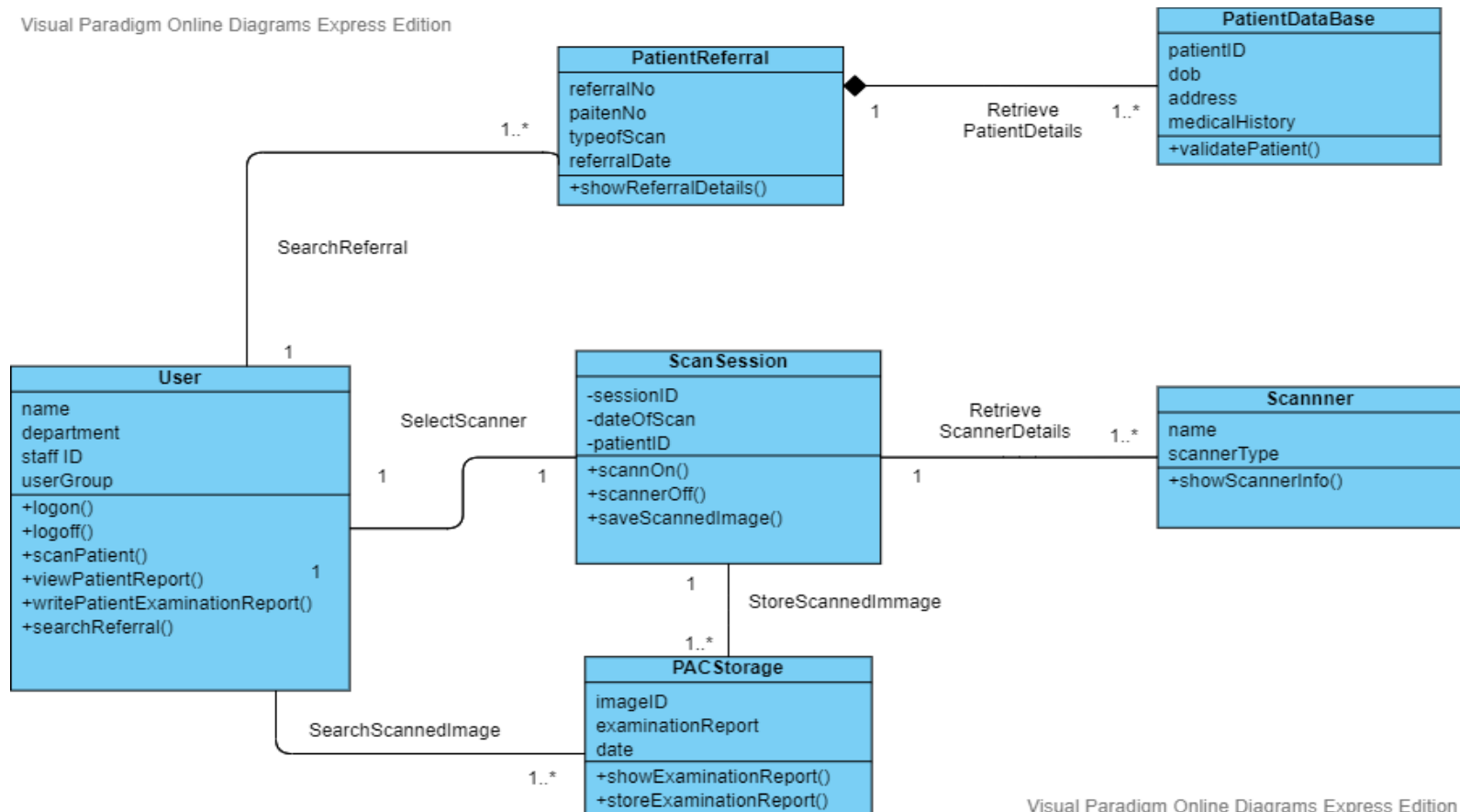
Discussion outline

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



Application to PACs

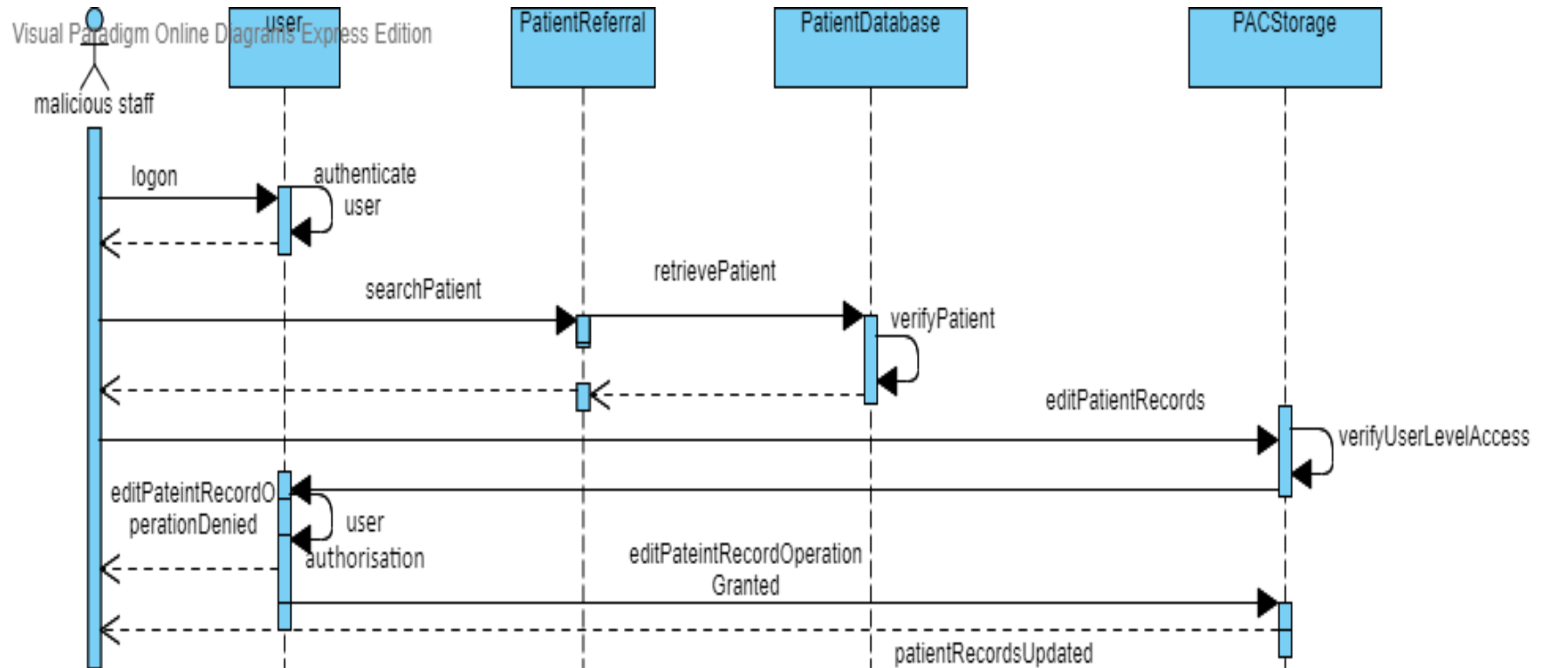
Visual Paradigm Online Diagrams Express Edition



Visual Paradigm Online Diagrams Express Edition



Application to PACs





Exercise

Implement PACs solution using OOP with Python and MySQL