

Term	Definition
User interface	The user interface is the point of human-computer interaction and communication in a device.
Input facilities	The facilities provided by a program that allow the user to enter data. These can include text prompts or text boxes.
Output facilities	The facilities provided by a program that allow data to be output to the user. These can include the display or a printer.
Data structures	A data structure is a specific way of organising data within memory so it can be processed efficiently.
Pseudo code	A notation resembling a simplified programming language, used in program design.
Validation	Validation is a process to check that input data is reasonable.
Verification	Verification is a process for checking data is correct.
Authentication	Authentication is the process of verifying the identity of a person or device.

This knowledge organiser will concentrate on designing a solution to a problem that will be coded in Python to support the skills needed to sit the Component 2 examination.

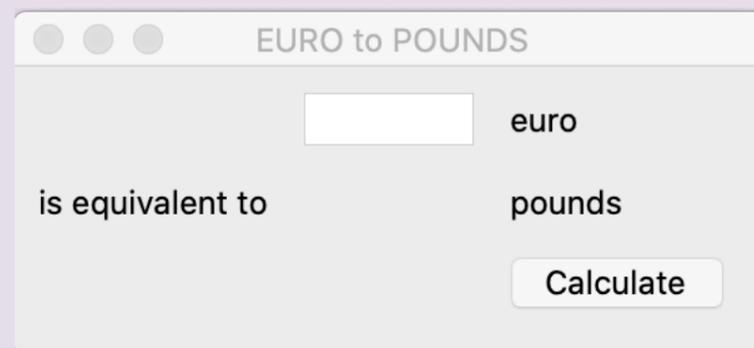
User interface

A user interface allows the user to interact with a program. It is important that an interface is well designed and is intuitive to use. There two options for creating a user interface in Python:

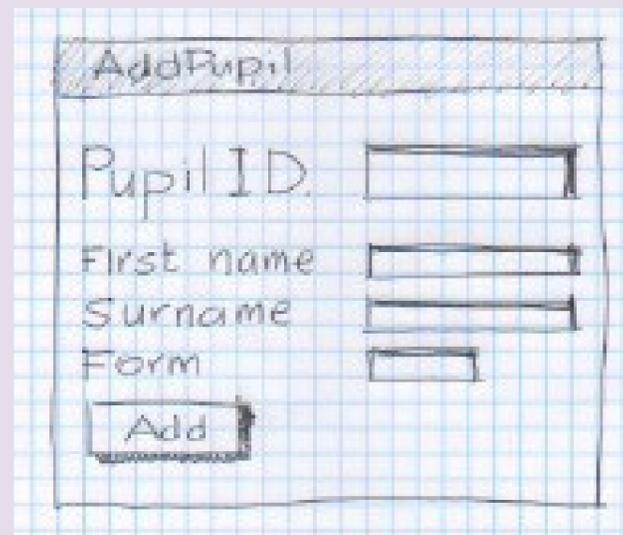
- Command line interface (CLI)

```
Enter your firstname please : Fred
Enter your surname please : Evans
Enter your address of where you live please : 10
High Street
Enter your title please : Mr
```

- Graphical user interface (GUI)



Component 2 will require a GUI. It is important that the layout of each screen is designed before the coding starts.



Data structures

The design for a program includes the consideration of data structures. To design a data structure a programmer needs to know what data will be collected and processed.

PupilID	First Name	Surname	Form
P0001	Fred	Smith	11W
P0002	Elsie	Davies	11B

In this case, the data about each pupil contains 4 items of data known as fields. All the information about one pupil is known as a record.

Data structures include information about each field including data type and field length. Data types include:

- Integer – whole numbers, 10, 23
- Real – numbers with decimal or fractional parts, 3.142, 99.9
- character – individual characters, M, F
- string – series of characters, Fred, Smith
- Boolean – TRUE or FALSE

Fieldname	Data type	Length	Example
PupilID	String	5	P0001
Firstname	String	20	Fred
Surname	String	20	Smith
House No	Integer		10
Postcode	String	8	CF62 6YX
Gender	Character	1	M
Attendance	Real		94.3

Validation and verification

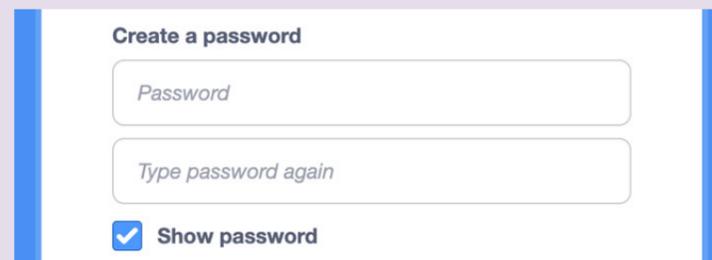
Validation is required to check that the data entered into a program is reasonable and sensible.

Validation rules are designed to check different data types. Validation rules include:

Rule	Description
Presence check	Checks to ensure that data has been entered into the user interface facility and has not been left blank.
Type check	Checks that data of the correct type has been entered into the field e.g. that a real number has not been entered into a real number field.
Format check	Checks that data is in the correct format, e.g. a post code (LL99 9LL) or a date of birth (DD MM YYYY)
Length Check	Checks that the data entered into a field is not longer than the set length of the field.
Range check	Checks that a value is within a given range e.g. month of the year is between 1 and 12.
Lookup Check	A check that allows the user to select from a list of options

Verification is the process that checks if data has been entered correctly. This can be done by eye, checking the data on the screen or printing out the data and checking it against the original data source.

Double entry is computer-based method of verification. This technique is often used when changing a password.



Data handling and processing

An important part of the design process is considering how the data entered will be manipulated and processed.

At the design stage data handling routines and all other processing are represented using pseudo code.

```
pupilID is string
firstName is string
surname is string

output "Please enter PupilID"
input pupilID
output "Please enter name"
input name

file = open("pupils.txt", "a")
file.write(pupilID + name "\n")
file.close()
```

Authentication

A common example of authentication is the using a username and password to logon to a website. The combination of username and password identifies the user and allows them to access the facilities of the website.

Some systems are designed to implement multi factor authentication. This means that users can only gain access after presenting at least two pieces of evidence (factors). Often this will involve entering a password and a code sent to a mobile phone or generated on a specialised device.

Authentication processes should also be designed:

```
Username as string
Password as string

output "enter username"
input username
output "enter password"
input password
```

Often websites use other methods such as a captcha which is known as a challenge response authentication. It consists of two parts, a randomly generated set of letters and or numbers.

