1. What is Python and what are its main features?

It is:

- Easy to code
- Free and open source
- An object oriented language
- A high level language (don't need to remember system architecture or manage memory)
- Extensible feature: python code can be written into C or C++
- It is portable: it can be run on windows, linux, unix or mac
- It is interpreted: it does not need time to be compiled
- It is dynamically typed: the variable type is decided when the programme is run, not in advance so type doesn't need to be specified

2. Discuss the difference between Python 2 and Python 3

Python 2.7 was a previous version of python that is no longer supported. In 2008, it was decided that the project would be 'sunset' in 2015 and people were encouraged to download/upgrade to python 3, but many people did not and so it was extended to 2020, when it was finally put to bed.

There have been many upgrades between 2 and 3 and changes to functions such as the print function which was a statement in python 2. Furthermore, numbers with no digits after a decimal are treated as integers and so division was not completed correctly e.g. 3/2 would = 1, whereas 3.0/2.0 would = 1.5

There were also further difference in syntax requirements, for example when raising exception as well as string storage – Unicode in python 3 and ASCII in python 2.

3. What is PEP 8?

It is a document that provides guidelines and best practices on writing python code – a style guide.

4. In computing / computer science what is a program?

A collection of instructions that can be executed by a computer to perform a specific task. They are written by people in a programming language.

5. In computing / computer science what is a process?

This is a program in execution

6. In computing / computer science what is cache?

A small amount of memory that is part of the CPU used to temporarily store instructions and data that will be reused by the CPU.

7. In computing / computer science what is a thread and what do we mean by multithreading?

A thread is a component of a process. It is the smallest sequence of programmed instructions that can be manage by a scheduler (scheduling is assigning resources to perform tasks). Multiple threads can exist within one process, executing at the same time and sharing resources like memory.

8. In computing / computer science what is concurrency and parallelism and what are the differences?

Concurrency is where 2+ tasks can start, run and complete in overlapping time periods but might not be running at the same time whereas parallelism is where tasks run at the same time.

9. What is GIL in Python and how does it work?

Global interpreter lock: stop race conditions and ensures safety of the thread by preventing multithreading from executing in memory at the same time.

10. What do these software development principles mean: DRY, KISS, BDUF

DRY: Don't repeat yourself: reduce repetition

KISS: Keep it simple stupid: don't overcomplicate – simplicity is key goal

BDUF: Big design up front: complete the design of a project first and then implement it

11. What is a Garbage Collector in Python and how does it work?

Garbage collection is the method by which 'dead objects' are removed from memory when their reference counter reaches 0.

e.g. x1 = apple = create a new object 'apple object 1' = 1 reference counter

x2 = x1 = refer to 'apple object 1' and create a new reference = 2 reference counter

x1 = None = remove a counter from 'apple object 1' = 1 reference counter

x2 = apple = create a new object 'apple object 2' and 'apple object 1' is set to 0

When the counter becomes 0 the object is removed from memory

12. How is memory managed in Python?

x = 10 - creates an integer object ‘10’ in memory with the variable x assigned to it.

y = x - would contain reference to the same object ‘10’

x = x + 1 would create a new object ‘11’ with the variable x assigned to it

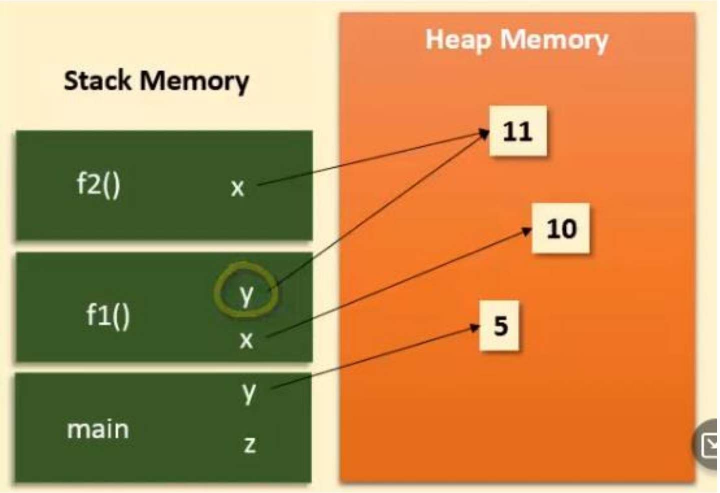
In python, memory utilisation is optimised by locating the same object reference to a new variable if the object already exists with the same value. Any variable can later be assigned to a new object (this is why python is dynamically typed).

In memory, processes are allocated memory by the operating system – this memory is divided into heap and stack. Methods and variables are executed in stack memory and objects are created in heap memory. Different stack frames with the same variable in them do not overwrite each other.

e.g.

```
def f2(x)
        x = x + 1
        return x

def f1(x):
        x = x*2
        y=f2(x)
        return y
y = 5
z = f1(y)
```



13. What is a Python module?

A module is a file that contains python definitions and statements and has the suffix .py

You can import modules into other modules or scripts using import <name of module>. A module can contain executable statements as well as function definitions.

It is also possible to give the module a nickname for further use for e.g.

```
import <name of module> as <nickname>
```

14. What is docstring in Python?

A docstring is a string literal that is placed directly after the definition of a function, method, class or module and describes what it does – it is a documentation string.

```
E.g. def square(n):
          “””Takes in a number n, returned the square of n””” ← docstring
          return n**2
```

The docstring can be printed e.g. print(square.__doc__)

Docstrings can also be printed for a module e.g.

```
import <module>
print (<module name>.__doc__)
```

15. What is pickling and unpickling in Python? Example usage.

Pickling is saving a python object into another form for e.g. binary and unpickling is turning that binary data back into an object. Pickling is used because the saved file is smaller and it is faster to load back in, especially if you are using it on large items (e.g. convert a large dataset into an object and then pickle)

e.g.

```
example = {“a”:”1”, “b”:”2”, “c”:”3”}
pickle_out = open(“pickle_file”, “wb”)
pickle.dump(example, pickle_out)
pickle_in = open(“pickle_file”, “rb”)
new_example = pickle.load(pickle_in)
print(example)
print(new_example)
```

16. What are the tools that help to find bugs or perform static analysis?

Pychecker and Pylint are static analysis tools that help find bugs in python.

Pychecker is an opensource tool for static analysis that detects the bugs from source code and warns about the style and complexity of the bug.

Pylint is highly configurable and acts like special programs to control warnings and errors, it is an extensive configuration file. Pylint is also an opensource tool for static code analysis. It looks for programming errors and is used for coding standards. It checks the length of each programming line, the variable names according to the project style and it can also be used as a standalone program. Furthermore, it can integrate with IDEs such as Pycharm, Spyder, Eclipse and Jupyter.

17. How are arguments passed in Python by value or by reference? Give an example.

By reference. If a change is made to what a parameter refers to within a function, then the change is reflected when the function is called. e.g.

```
fruit_basket = {“apple”:  1, “banana”: 2, “grapes”: 5}
def example (fruit_basket):
        new = {“melon”: 2, “oranges”: 7}
        fruit_basket.update(new)
        print(“Inside the function”, fruit_basket)
        return
example(fruit_basket)
print(“Outside the function”, fruit_basket)
```

Output: Inside the function {“apple”: 1, “banana”: 2, “grapes”: 5, “melon”: 2, “oranges”: 7}

Output: Outside the function {“apple”: 1, “banana”: 2, “grapes”: 5, “melon”: 2, “oranges”: 7}

18. What are Dictionary and List comprehensions in Python? Provide examples.

List comprehensions are a more elegant way of creating lists than a for loop (and can also create lists from lists).

e.g. a for loop:

```
for i in range(5):
        print(i)
```

Output:

```
0

1

2

3

4
```

e.g. a list comprehension

```
x = [i for i in range(10)]
```

Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

e.g. a conditional statement can also be added

```
x = [i for i in range(10) if i >5]
```

Output: [6, 7, 8, 9]

Dictionary comprehensions are similar to list comprehensions:

e.g.

dict([(i, i+10) for i in range(4)]) is equivalent to {i: i+10 for i in range(4)}

Output: {0: 10, 1: 11, 2: 12, 3: 13}

An if statement can be used to filter out values to create a new dictionary:

```
{i: i+10 for i in range(10) if i > 5}
```

Output: {6: 16, 7: 17, 8: 18, 9: 19}

19. What is namespace in Python?

Namespaces are the names assigned to objects in a python project. There are four types:

- Built-in: these are the built-in objects and can be listed using dir(__builtins__)
- Global: names defined at the main program level and also imported modules
- Enclosing: the namespace for a function that has another function within it
- Local: the namespace for the function within another function

e.g.

```
def f():
    print("Start f()")
    def g():
        print("Start g()")
        print("End g()")
        return
    g()
    print("End f()")
    return
f()
```

Output:

```
Start f()
Start g()
End g()
End f()
```

f() is the enclosing namespace and g() is the local namespace

20.What is pass in Python?

Pass is a keyword that is used as a placeholder for future code. E.g.

```
def addition(num1, num2):
            pass


def subtraction(num1, num2):
            print(“Sub: “, num1 – num2)


addition(3, 2)
subtraction(3, 2)
```

Output: Sub: 1

21. What is unit test in Python?

This is a software testing method where individual units of code are tested to see if they are fit for use. It determines the quality of the code.

22. In Python what is slicing?

The slice function returns only part of an object. It is specified as start:end:step E.g.

```
x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print(x[3::2])
```

Output: [3, 5, 7, 9]

23.What is a negative index in Python?

This is where the index starts at the end, the first item of a negative index (the last element) is -1

24.How can the ternary operators be used in python? Give an example.

Ternary operators are also called conditional expressions – they evaluate something based on a condition being True or False. E.g.

```
a, b = 10, 20
min = a if a < b else b
print(min)
```

Output: 10

Tuples, dictionaries and lambda can also be used to carry out these expressions:

Tuple: print((b, a) [a < b])

Dictionary: print({True: a, False: b} [a < b])

Lambda: print((lambda: b, lambda: a)[a < b]())

25.What does this mean: *args, **kwargs? And why would we use it?

*args = non-keyword argument

*kwargs = keyword argument

Both of these allow the passing of multiple arguments to a function.

*args e.g.

```
def example(*argv):
            for arg in argv:
                        print(arg)
example("Hello", "my", "name", "is", "Jo")
```

Output:

```
Hello
my
name
is
Jo
```

argv can be any name with a * in front of it, e.g. *integers, *fruits, *anything

**kwargs e.g.

```
def example(**kwargs):
            for kwarg in kwargs.items():
                        print(kwarg)
example(name = "Jo", age=34)
```

Output:

```
('name', 'Jo')
('age', 34)
```

26. How are range and xrange different from one another?

These functions are both used to generate a sequence of numbers, however xrange() is only used in python 2 whereas range() is used in python 3.

range() returns a list and xrange() returns the type xrange.

range() uses a lot more memory → range(1,10000) uses 800064 whereas xrange(1,10000) uses 40

27. What is Flask and what can we use it for?

Flask is a third-party library used for developing web applications.

28.What are clustered and non-clustered index in a relational database?

A relational database is a collection of data items with pre-defined relationships between them which are organised as a set of tables with columns and rows. Identifiers can be marked with a primary key and the tables are related using foreign keys.

Indexes are used to speed up query processes, there are two kinds: clustered and non-clustered.

A clustered index defines the order that data is physically stored in a table – in SQL the primary key creates a clustered index on the column it is added to, there can only be one clustered index per table because the rows of data can only be stored in one order. A clustered index is like a phone book that is ordered by name and has a name index at the top of the page that says the range of surnames on those pages.

A non-clustered index is more like the contents page in the back of a book – it is a list of information that has pointers to tell where the information is stored. A non-clustered index is not stored in the same place as the data it points to.

If indexes are not used then the data has to be searched using a table-scan which scans every row to look for the information and will continue to the end because it doesn't know if it will appear again!

29.What is a 'deadlock' a relational database?

A database lock is used to "lock" data in a database so that only one database user/session can update that data – so the exact same piece of data cannot be updated by more than one user at the same time.

A deadlock is where two or more transactions are waiting for each other to give up locks.

e.g. Transaction A has a lock on rows in table 1 and needs to update rows in table 2 to finish, however Transaction B has a lock on the rows in table 2 and needs to update the rows that are locked by Transaction A in table 1. Neither Transaction A or B can complete and are deadlocked – one of the transactions must be aborted to allow activity to continue.

30.What is a 'livelock' a relational database?

A livelock is where a request for an exclusive lock is continuously denied because requests keep overlapping with each other, then adapt, change their status and are prevented from completing a task. The analogy of two people trying to get around each other in a hallway who keep stepping to the same side ("dancing") is equivalent to these requests for locks blocking each other. In SQL a livelock occurs when read transactions prevent write transactions and cause them to wait indefinitely.