1. *What is shallow copy and deep copy:* ***Cisco, Calsoft.***

Ans: *Shallow copy and deep copy functions can be defined as below:*

1. *A shallow copy constructs a new compound object and then inserts references into it to the objects found in the original.*
2. *A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original.*
3. *With a shallow copy, any object pointed to by the source is pointed to by the destination (so that no referenced objects are copied).*
4. *With a deep copy, any object pointed to by the source is copied into the destination (so there will now be 2 of each referenced object).*

Shallow Copy: *Simply makes a copy of the reference to A into B. Think about it as a copy of A's Address. So, the addresses of A and B will be the same i.e. they will be pointing to the same memory location i.e. data contents.*
Deep copy: *Simply makes a copy of all the members of A, allocates memory in a different location for B and then assigns the copied members to B to achieve deep copy. In this way, if A becomes non-existant B is still valid in the memory.*
'ShallowCopy' points to the same location in memory as 'Source' does.
'DeepCopy' points to a different location in memory, but the contents are the same.
Shallow Copy is similar to Call By Reference and a Deep Copy is similar to Call By Value.
Shallow copy *involves copying the contents of one object into another instance of the same class.*
Deep copy *involves using the contents of one object to create another instance of the same class.*

2. *What is repr function and what does it return?:* ***Cisco.***

Ans: *Python can convert any value to a string by making use of two functions repr() or str(). The str() function returns representations of values which are human-readable, while repr() generates representations which can be read by the interpreter. repr() returns a machine-readable representation of values, suitable for an exec command.*

3. *Difference between overloading and overriding:* ***IBM***

Ans: *overloading deals with the notion of having two or more methods(functions) in the same class with the same name but different arguments.*
overriding means having two methods with the same arguments, but different implementation.

4. *What is a lambda function and what does it do? :* ***Cisco.***

Ans: *Lambda is a single expression anonymous function often used as inline function. Small anonymous functions can be created with the lambda keyword. Lambda functions can be used wherever function objects are required. They are syntactically restricted to a single expression. It takes general form as:*
lambda arg1 arg2 ... : expression where args can be used.

5. *Differences between tcp and udp:* ***McAfee, Wipro, Novell, IBM***

   Ans: *http://www.diffen.com/difference/TCP_vs_UDP*

6. *What is a heap memory?* ***Graphene, Wipro, etc.***

   Ans: *The heap memory segment is an area of memory used for dynamic allocations,* ***heap*** *is the portion of memory where dynamically allocated memory resides (i.e. memory allocated via malloc). Memory allocated from the heap will remain allocated until one of the following occurs:*
   1. *The memory is free'd*
   2. *The program terminates.*

7. *What are decorators in python?* ***Paxterra, Cisco, Graphene, Calsoft.***

   Ans: *A decorator is a function whose primary purpose is to wrap another function or class.*
   The primary purpose of this wrapping is to alter or enhance the behavior of the object being wrapped. A decorator is just a callable that takes a function as an argument and returns a replacement function. Decorators dynamically alter the functionality of a function, method, or class without having to directly use subclasses or change the source code of the function being decorated. The @ symbol applies a decorator to a function. We can wrap a function in a decorator by pre-pending the function definition with a decorator name and the @ symbol.

8. *What are generators?*

   Ans: Generators *are simple functions which return an iterable set of items. A generator is a function that produces a sequence of results instead of a single value.*

9. *What is the difference between range and xrange functions?* ***Paxterra, Cisco, Graphene, Calsoft.***

   Ans: *The differences are as follows:*
   - *range returns a list object and xrange returns an xrange object.*
   - *It means that range generates a static list at run-time and xrange creates the values as you need them with a special technique called yielding which is a technique used with the generators.*
   - *xrange returns an iterator and only keeps one number in memory at a time.*

- *range keeps the entire list of numbers in memory.* range returns a static list at runtime. xrange returns a generator object from which values are generated as and when required.

10. *How is memory managed in python?* ***Paxterra, Cisco, Graphene, Calsoft.***

- *Memory management in Python involves a private heap containing all Python objects and data structures. Interpreter takes care of Python heap and that the programmer has no access to it.*
- *The allocation of heap space for Python objects is done by Python memory manager.*
- *Python also has a build-in garbage collector which recycles all the unused memory. When an object is no longer referenced by the program, the heap space it occupies can be freed. The garbage collector determines objects which are no longer referenced by the program frees the occupied memory and make it available to the heap space.*
- *The gc module defines functions to enable /disable garbage collector: gc.enable() -Enables automatic garbage collection. gc.disable() - Disables automatic garbage collection.*

11. *What is garbage collection?* ***Paxterra, Cisco, Graphene, Calsoft.***

Ans: *Garbage collection is a form of automatic **memory management**. The garbage collector, or just collector, attempts to reclaim **garbage**, or memory occupied by **objects** that are no longer in use by the **program**.* ***OR*** *Garbage collection is the systematic recovery of pooled computer **storage** that is being used by a program when that program no longer needs it. This frees the storage for use by other programs or processes within a program.*

12. *What is encapsulation?*

Ans: ***Encapsulation*** *is the packing of data and functions into a single component. The features of* ***encapsulation*** *are supported using classes in most object-oriented programming languages, although other alternatives also exist.*

13. *What are the different features of python:* ***Wipro, Mphasis***

Ans: http://www.swaroopch.com/notes/python/#_features_of_python

14. *What is the difference between list, tuple, dictionary and a set:* ***Wipro, HP, SanDisk, McAfee, Dell***

Ans: *A dictionary is an associative array or hash table that contains objects indexed by keys.*

Lists are sequences of objects.

http://www.dreamincode.net/forums/topic/82368-data-structures-in-python/

15. *What is the use of __init__.py file in python:* ***SanDisk***

Ans: *Files named __init__.py are used to mark directories on disk as a Python package directories. If you have the files mydir/spam/__init__.py*

mydir/spam/module.py
and ***mydir*** *is on your path, you can import the code in module.py as:*

import spam.module
or

from spam import module
If you remove the __init__.py file, Python will no longer look for submodules inside that directory, so attempts to import the module will fail.

The __init__.py file is usually empty, but can be used to export selected portions of the package under more convenient names, hold convenience functions, etc. Given the example above, the contents of the __init__ module can be accessed as:

import spam.

16. *How tcp/ip works:* ***Novell, Wipro***

Ans: *TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol (TCP), manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol (IP), handles the address part of each packet so that it gets to the right destination. TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer.*

17. *Explain the TCP/IP protocol suite:* ***Novell, McAfee.***

Ans. *The TCP/IP protocol maps to a four layer conceptual model: application, transport, Internet and network interface. This model is referred to as the Internet Protocol Suite or the ARPA model.*
NetworkInterface
The network interface layer is the equivalent of the OSI physical and data link layers as it defines the host's connection to the network. This layer comprises the hardware and software involved in the interchange of frames between

computers. The technologies used can be LAN-based (e.g. Ethernet) or WAN-based (e.g. ISDN).

InternetLayer

The network layer uses the protocols to ensure the delivery of packets. These are described below:

IP(InternetProtocol)

IP is the protocol responsible for addressing and routing packets (on the basis of routing algorithms) between networks. It ensures the packets reach the correct destination network.

ARP

The Address Resolution Protocol (ARP) is responsible for obtaining hardware addresses and matching them to their IP address when the destination computer is on the same network.

ICMP

The Internet Control Management Protocol (ICMP) is responsible to report errors and send messages about the delivery of a packet. It can also be used to test TCP/IP networks. Two examples of ICMP messages include:

- ***Destination unreachable*** *- when a router cannot locate the destination.*
- ***Time exceeded*** *- when the Time To Live (TTL) of a packet reaches zero.*

TransportLayer:

In the Open Systems Interconnection (OSI) communications model, the Transport layer ensures the reliable arrival of messages and provides error checking mechanisms and data flow controls. The Transport layer provides services for both "connection-mode" transmissions and for "connectionless-mode" transmissions. For connection-mode transmissions, a transmission may be sent in the form of packets that need to be reconstructed into a complete message at the other end. The Transport layer provides communication between the source and destination computers, and breaks application layer information into packets. TCP/IP provides two methods of data delivery:

- ***Connection-orientated*** *delivery using TCP*
- ***Connectionless*** *delivery using UDP.*

Application Layer

The Application layer is the layer at which many TCP/IP services (high level protocols) can be run (such as FTP, HTTP and SMTP). Two application programming interfaces (APIs) are commonly used within the TCP/IP environment:

- *sockets*
- *NetBIOS*

18. *Connect to a remote machine and execute a command using python:* ***Mindteck, McAfee, Wipro***

Ans:

19. *Linux Boot Process:* ***Accenture***

Ans: *Refer the print out.*

20. *RAID Terminologies:* ***Novell, Wipro***

Ans: *Refer the print out.*

21. *Differences between paging and swapping:* ***Accenture****.*

Ans: *The differences between paging and swapping can be explained as follows:*
Paging: *Paging is a memory management method used by operating systems. Paging allows the main memory to use data that is residing on a secondary storage device. These data are stored in the secondary storage device as blocks of same size called pages. Paging allows the operating system to use data that will not fit in to the main memory. When a program tries to access a page, first the page table is checked to see whether that page is on the main memory. Page table holds details about where the pages are stored. If it is not in the main memory, it is called a page fault. Operating system is responsible for handling page faults without showing it to the program. The operating system first finds where that particular page is stored in the secondary storage and then brings it in to an empty page frame in the main memory. Then it updates the page table to indicate that the new data is in the main memory and returns the control back to the program that initially requested the page.*
Swapping: *Swapping is the process of moving all the segments belonging to a process between the main memory and a secondary storage device. Swapping occurs under heavier work loads. Operating system kernel would move all the memory segments belonging to a process in to an area called swap area. When selecting a process for swapping, the operating system will select a process that will not become active for a while. When the main memory has enough space to hold the process, it will be transferred back in to the main memory from the swap space so that its execution could be continued.*

What is the difference between Paging and Swapping?
In paging, blocks of equal size (called pages) are transferred between the main memory and a secondary storage device, while in swapping, all the segments belonging to a process will be moved back and forth between the main memory and a secondary storage device. Since paging allows moving pages (it could be a part of the address space of a process), it is more flexible than swapping. Since, paging only moves pages (unlike swapping, which move a whole process), paging would allow more processes to reside on the main memory at the same time, when compared with a swapping system. Swapping is more suitable when running heavier workloads.