Siemens Limited is a technology firm specialising in the industry, infrastructure, digital transformation, transportation, and electrical power transmission and generation. It is Siemens AG's flagship listed firm in India. The corporate structure of the organisation is well-suited to satisfying industry demands for increased efficiency, quality, flexibility, and speed. The product-based company is well-positioned to assist the country in achieving long-term success thanks to its diverse portfolio, market-oriented organisational structure, global technology leadership, and strong local competency.

In almost every country on the planet, there are possibilities to join Siemens. Engineering, Manufacturing, Research & Development, IT, Sales & Marketing, and Operations are all areas where they're seeking new talent.

Siemens Recruitment Process

1. Interview Process

The interview process at Siemens usually consists of 5 rounds. Each of these rounds is an elimination round. The interview process aims at testing your technical as well as verbal skills. The following are the interview rounds mentioned in detail.

2. Interview Rounds

1. **Resume Screening:** In this stage, the resume screening takes place out of which a proportion of individuals are shortlisted.
2. **Online Technical Assessment:** This section consists of Programming Principles and Coding. There are about 30 questions based on programming concepts.
3. **Online Aptitude Assessment:** This section contains Quantitative, Verbal, and Logical Reasoning questions. The questions are simple, but you must be thorough with your answers. It must be completed in a given amount of time and there will be paragraph-based questions in both the verbal and LR sections.
4. **Technical Interview Round:** For this section, the basic questions start with C/C++ principles. Your projects can be discussed here. What technologies did you employ, why did you use them, and how did you use them? Apart from it DSA and algorithms is the main part of this interview and questions related to these are asked in the interview. Other topics that can be asked here could be based on computer fundamentals like OOPS, Computer networks, Operating systems etc. There can also be case studies for which you would have to write code for specific instances. For this round, you must be technically proficient. There can be multiple rounds for the technical round. Also, for experienced candidates, some of these rounds can be skipped.
5. **HR Interview Round:** This round includes an overview of a person, apart from the things mentioned in the resume. Questions like - 'How do you stay motivated at work?' and 'Why SIEMENS?' can be asked. Discussions about academic history, internship, family history, and extracurricular activities are included in this round. Some additional questions could include - ‘When working on a project, how do you come up with strategies?’.

Siemens Technical Interview Questions: Freshers and Experienced

1. What is a web application?

In contrast to computer-based software applications that run locally on the operating system (OS) of the device, a web application (or web app) is application software that operates on a web server. The user uses a web browser with an active network connection to access web apps. These apps are built on a client-server architecture, in which the user ("client") receives services from an off-site server hosted by a third party. Webmail, online retail purchases, online banking, are all examples of regularly used web applications.

2. What are the advantages and disadvantages of DHCP?

DHCP stands for Dynamic Host Configuration Protocol, and it is a client-server architecture.

The following are the **advantages or benefits** of DHCP:

- DHCP is simple to set up and assign IP addresses to request clients automatically. As a result, the time it takes to manually configure IP addresses can be decreased.
- There are no additional expenditures associated with the deployment.
- Duplicate or erroneous IP address assignment is avoided. As a result, there are no IP address disputes.
- It makes network administration easier.
- It supports a variety of scopes, such as multicast and super scopes.
- It benefits mobile users greatly because correct configuration parameters are received immediately from the new network.

The following are DHCP's **drawbacks or downsides**:

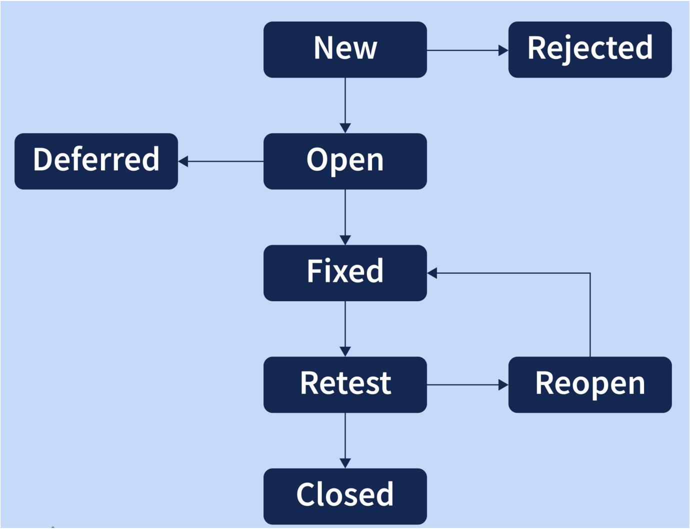
- In networks with only one configured DHCP server, the DHCP server might be a single point of failure.
- Because DHCP packets cannot cross routers, a relay agent is required to ensure that the DHCP server handles all leases on both network segments. Relay agents accept broadcast DHCP packets and forward them to the DHCP server as unicast packets. In this case, the relay agent must be configured with the DHCP server's IP address.
- Security: Because the DHCP server lacks a secure means for client authentication, it can get unauthorised access to IP addresses by submitting credentials such as DHCP client identifiers.
- When a new IP address is assigned, the machine name does not change.

3. Explain Bug Life Cycle.

The bug life cycle, also known as the defect life cycle, is a process by which a defect progresses through many phases throughout its lifetime. This lifetime begins when a tester reports a bug and ends when a tester determines that the problem has been resolved and will not recur.

In a defect workflow, the following defects states can occur.

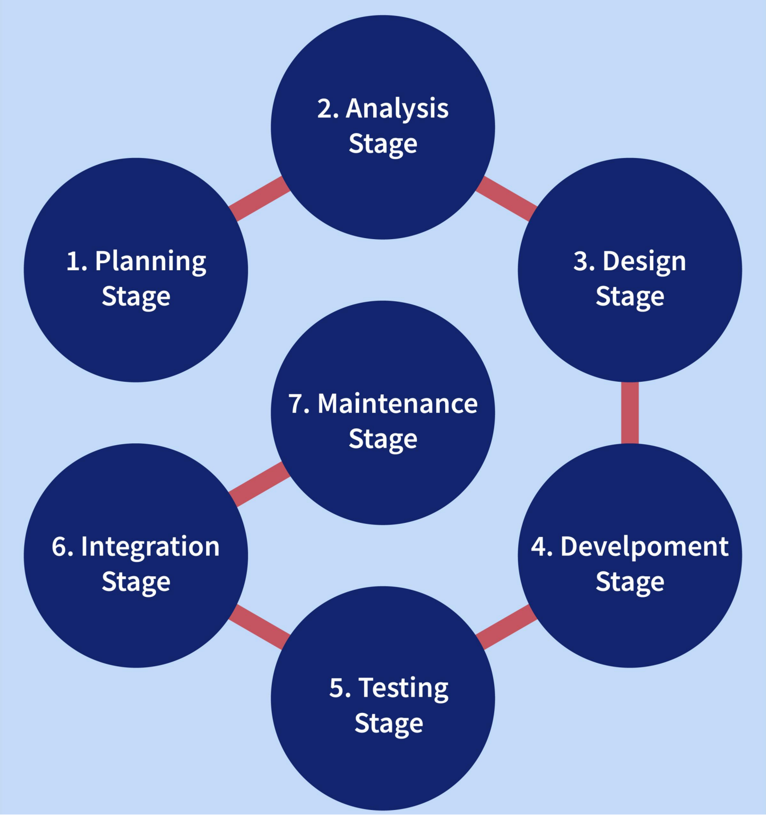
- **New:** When a new defect is first documented, it is referred to as "new" and given the status of New.
- **Assigned:** Once the tester has filed the bug, the tester's lead accepts it and assigns it to the programming team.
- **Open:** The developer begins analyzing and working on the defect.
- **Fixed:** A developer can mark an issue as "Fixed" once he or she has made a necessary code modification and verified it.
- **Pending Retest:** Once the defect is fixed, the developer gives the tester a specific code to retest the code. The status given is "pending retest" because the software testing is still waiting from the testers' end.
- **Retest:** At this level, the tester retests the code to see if the defect has been fixed by the developer and changes the status to "Re-test."
- **Reopen:** The tester updates the status to "reopened" if the bug continues after the developer has solved it. The bug goes through the life cycle once more.
- **Verified:** The tester re-tests the bug after the developer has solved it. If no bugs are found in the software, the bug is fixed, and the status is changed to "verified."
- **Closed:** When a bug is no longer present, the tester marks it as "Closed."



4. What are SDLC phases?

The SDLC (system development life cycle) is a project management model. It specifies the several stages that must be completed in order to take a project from conception to deployment and afterwards maintenance.

The 7 Stages of the System Development Life Cycle:



- **Planning Stage:** The planning stage (also known as the feasibility stage) is the time when developers start thinking about the next project. It helps in understanding the definition of the problem and scope of any existing systems, as well as the determination of the new system objectives.
- **Feasibility or Requirements of Analysis Stage:** Gathering all of the specific details required for a new system, as well as defining the first prototype concepts, is part of the analysis step.
- **Design and Prototyping Stage:** The design stage is required before moving on to the primary developer stage. Developers will begin by outlining the overall application's characteristics, as well as individual aspects such as User interfaces, System interfaces, Network requirements, Databases.
- **Software Development Stage:** The development stage is when programmers write code and build the application based on the design papers and specifications that were created earlier.
- **Software Testing Stage:** Developers will go over their software with a fine-tooth comb during the testing stage, identifying any flaws or defects that need to be recorded, corrected, and retested.
- **Implementation and Integration:** The general design for the app will come together after testing. Developer efforts will integrate different modules or designs into the source code, usually by using training environments to find more faults or defects.
- **Operations and Maintenance Stage:** This can include resolving new issues that arise as a result of user reports or dealing with leftover bugs that were not able to be corrected before launch.

5. What are the types of kernel objects?

The following are the several types of kernel objects:

- Access token objects
- Event objects
- File objects
- File-mapping objects
- I/O completion port objects
- Job objects
- Mailslot objects
- Mutex objects
- Pipe objects
- Process objects
- Semaphore objects
- Thread objects
- Waitable timer objects

6. What is the use of pointers?

Pointers are used to store and maintain the addresses of memory blocks that are dynamically allocated. Data objects or arrays of objects are stored in these blocks. The heap or free store is a memory space in most structured and object-oriented languages from which objects are dynamically allocated.

Example in C:

```
#include <stdio.h>
void printpointer()
{
 int var = 10;
 int *pt;
 pt = &var;
 printf("The address is: %p \n",pt);
 printf("The value is: %d \n", *pt);
}
int main()
{
 printpointer();
}
```

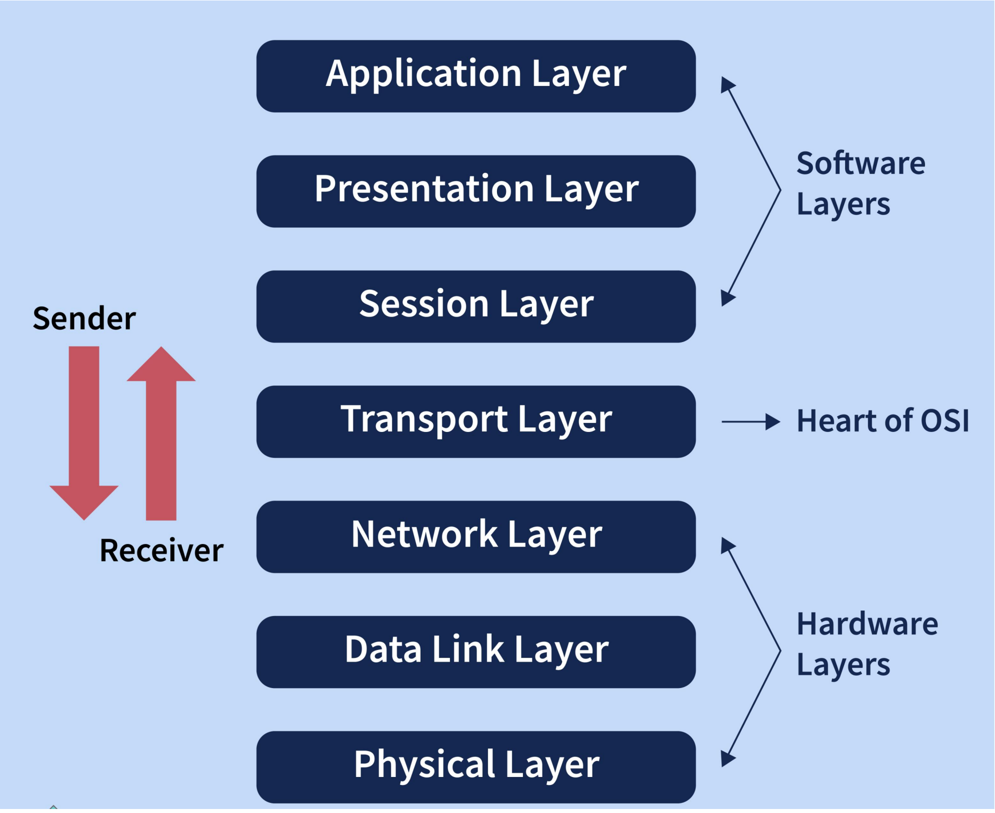
7. What is meant by stack and queue?

- A **stack** is a linear data structure in which elements can only be added or removed from the top side of the list. The LIFO (Last In First Out) principle dictates that the element put last is the first to come out of a stack. Pushing an element into a stack is referred to as a push operation while removing an element from a stack is referred to as a pop operation. With a pointer called top, we always maintain track of the last entry in the list in the stack.
- A **queue** is a linear data structure in which elements can only be inserted from one side of the list, called the back, and only deleted from the other side, called the front. The FIFO (First In First Out) principle governs the queue data structure, which means that the element placed first in the

list is the first one withdrawn from the list. Enqueue operations are used to add items to a queue, and dequeue operations are used to remove items from a queue.

8. What are the different types of OSI layers?

Open Systems Interconnection (OSI) is an acronym for Open Systems Interconnection. It was created in 1984 by ISO or the International Organization for Standardization. It is a seven-layer architecture, with each layer performing distinct functions. These seven layers work together to send data from one person to another across the globe.



- **Physical Layer:** The physical layer is the lowest layer in the OSI reference model. It is in charge of establishing a physical connection between the devices. Bits of information are stored in the physical layer. It is in charge of sending individual bits from one node to another. When this layer receives data, it converts the signal received into 0s and 1s and sends them to the Data Link layer, which reassembles the frame.
- **Data Link Layer:** The data link layer is in charge of message transport from node to node. The major purpose of this layer is to ensure that data transfers from one node to another through the physical layer are error-free. There are two sublayers in the Data Link Layer:
  - Media Access Control (MAC)
  - Logical Link Control (LLC)
- **Network Layer:** The network layer is responsible for data transmission between hosts that are connected to various networks. It also handles packet routing, which is the choosing of the

shortest path to send a packet from a large number of options. The network layer places the IP addresses of the sender and receiver in the header.

- **Transport Layer:** The application layer receives services from the transport layer, while the network layer receives services from the transport layer. Segments are the units of data in the transport layer. It is in charge of the full message's delivery from beginning to end. If an error is detected, the transport layer acknowledges the successful data transmission and re-transmits the data.
- **Session Layer:** This layer is in charge of establishing connections, maintaining sessions, authenticating users, and ensuring security.
- **Presentation Layer:** The data from the application layer is retrieved and processed here so that it may be transmitted across the network in the proper format.
- **Application Layer:** The Application layer, which is implemented by network applications, is at the very top of the OSI Reference Model stack of layers. These programs generate the data that must be sent across the network. This layer also acts as a window for application services to connect to the network and show the information they receive to the user.

9. What is the default port for the MySQL server?

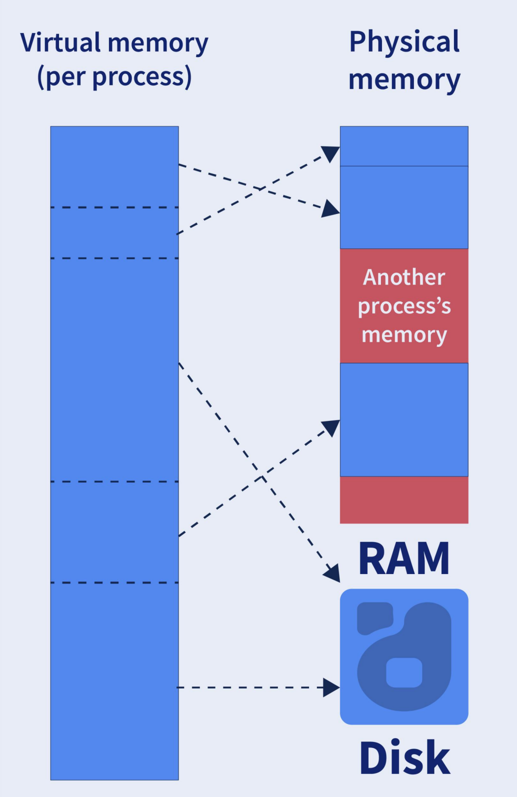
The MySQL client, MySQL Connectors, and utilities like mysqldump and mysqlpump all use port 3306 as the default port for the old MySQL protocol (port).

10. What is a virtual memory?

A segment of memory that is produced momentarily on the storage device is referred to as virtual memory. It occurs when a computer's RAM gets depleted as a result of multiple processes executing at the same time.

A part of the storage disc is made accessible for usage as RAM by the operating system. Because processor power is consumed by moving data around rather than performing instructions, virtual memory is substantially slower than main memory. The operating systems guide shows how the operating system handles memory. When the computer needs to use virtual memory, latency increases. The operating system employs swapping to transport data between RAM and virtual memory. Data from processes that aren't in use right now is moved out of RAM and stored in virtual memory by the operating system. When the process is needed again, it copies the data back into RAM.

As transferring to a hard disc takes far longer than reading and writing RAM, using virtual memory slows down the machine.



11. What is a parent/child selector?

All elements that are a direct child of the specified element are selected using the ("parent > child") selection.

Parameters:

- **Parent:** This specifies the parent element to be selected.
- **Child:** This specifies the direct child element (of the specified parent element) to be selected.

It chooses and returns all of the parent element's direct children.

Here is an example of how to use the jQuery parent > child selector:

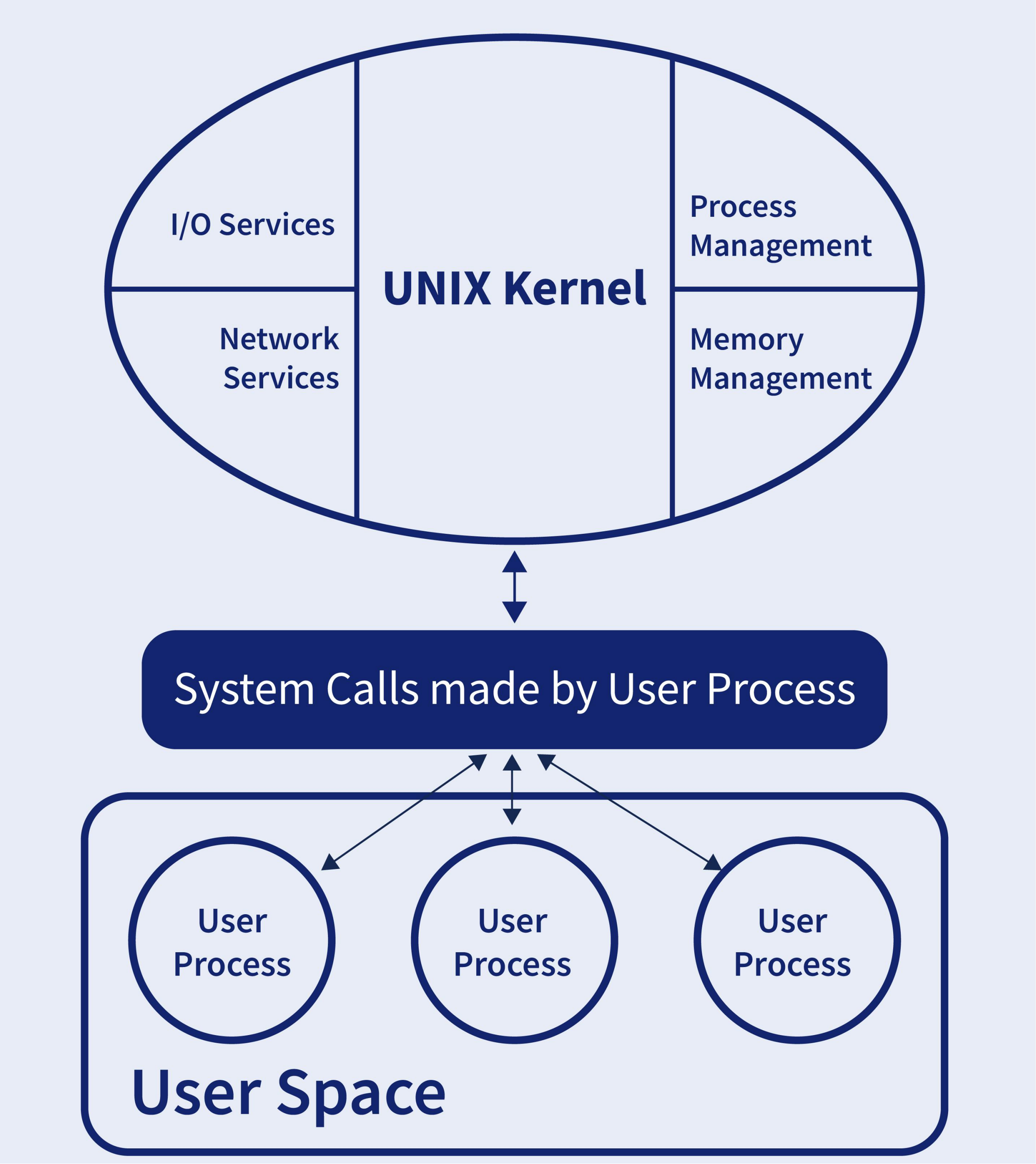
```
<html>
<body>
<h2>Sample Demo</h2>
<div>
<p>First line.</p>
<span>Middle line. (span outside of p element.) </span>
<p>Last line. </p>
</div><br>
<div>
<p>First line.</p>
```

```
<p>Middle line. <span>span inside p element.</span></p>
<p>Last line. </p>
</div>
</body>
</html>
```

12. Explain the Unix kernel.

The operating system's central core is the UNIX kernel. It connects to the hardware devices, as well as to the processor, memory, and I/O management. Users' requests are managed by the kernel using system calls that move the process from user space to kernel space.

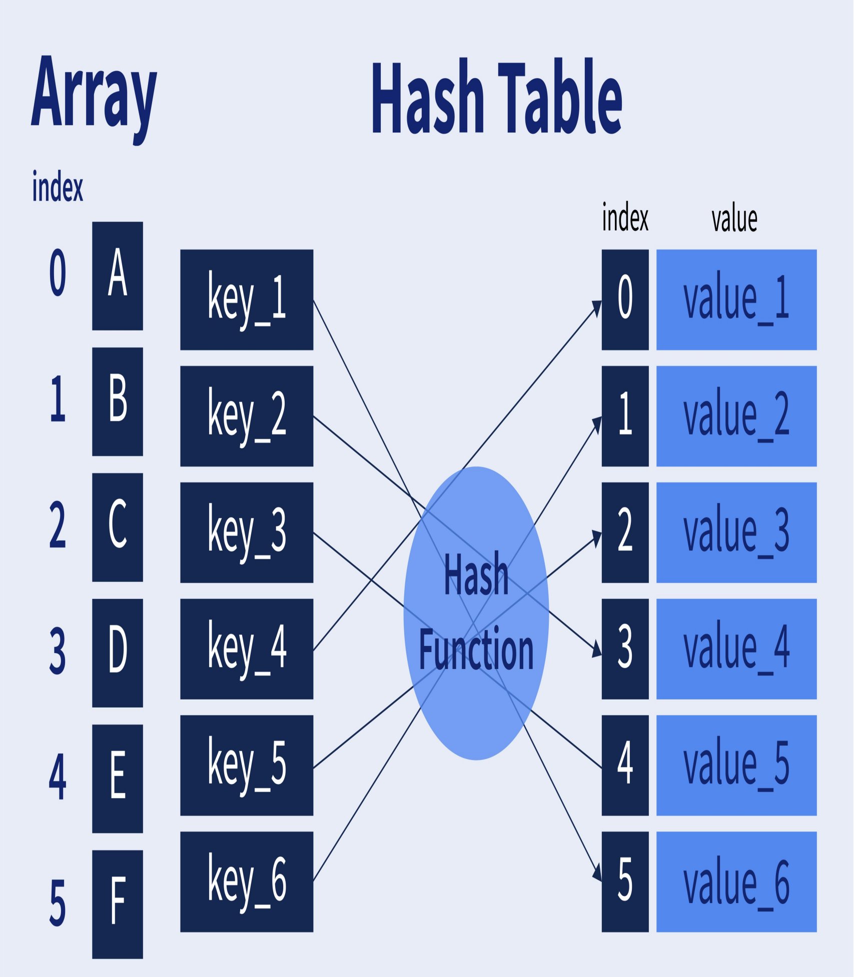
A context switch occurs every time a user process performs a system call, such as read(), fork(), exec(), open(), and so on. A context switch is a mechanism that allows a process to change its state. The process can either be blocked until the system call is completed, or it can continue and be told of the completion of the system call via a signal (nonblocking).



13. What is the difference between array and hash table?

An array has a set number of memory regions at first. It should store a preset type at each site (e.g. an integer). It could have repeated values or not.

Hash is implemented as a set, on the other hand. It's built around an array. A hash function determines the position of an item in an array. There should be no duplicates in a hash. A collision occurs when duplicates occur, and there are several techniques for resolving this (e.g. chain duplicates, rehash and so on).



14. What is marshalling?

Marshalling is the process of converting an object's memory representation into a format that can be stored or transmitted to other software applications. Marshalling converts an object into serialised form, allowing communication between remote objects.

Serialization and marshalling are two terms that are often used interchangeably. The objective of marshalling is to have the same object that is present in one operating programme, to be present in another running programme, i.e. object on a client to be transmitted to and present on the server. For example, serialization does not always have this intention because it is simply concerned with translating data into a stream of bytes.

15. What are system calls?

A system call is a method through which a computer programme asks a service from the kernel of the operating system on which it is running. A system call is a method of interacting with the operating system via programmes. When a computer software makes a request to the kernel of the operating system, it is called a system call. The operating system's services are provided to user programmes via the Application Program Interface (API). It acts as a link between a process and the operating system, allowing user-level programmes to request operating system services. The kernel system can only be accessed using system calls. System calls are required for any programmes that use resources.

16. What daemon is responsible for tracking events on your system?

The daemon 'syslogd' is in charge of keeping track of system data and storing it in particular log files.

17. Describe how you would reverse a singly linked list.

The task is to reverse a **linked list** given a pointer to the head node. By modifying the linkages between nodes, we can reverse the list.

Iterative method:

- Set up three-pointers. prev is set to NULL, curr is set to head, and next is set to NULL.
- Go through the linked list one by one. Do the following in a loop.

```
// Before changing next of current,
// store next node
next = curr->next
// Now change next of current
// This is where actual reversing happens
curr->next = prev
// Move prev and curr one step forward
prev = curr
curr = next
```

18. What is the difference between Manual and Automation Testing.

There are some significant distinctions between automated and manual testing. Manual testing involves a human performing tests step by step without the use of test scripts. Test automation frameworks, as well as other tools and technologies, are used to automate the execution of tests in automated testing. **Learn More**.

Manual Testing	Automation Testing
Manual testing takes time and requires human resources.	Because automated testing is carried out by software tools, it is much faster than manual testing.
Manual testing is only feasible when the test	When test cases are run frequently over a

Manual Testing	Automation Testing
cases are executed only once or twice, with no need for frequent repetition.	lengthy period of time, automated testing is a viable alternative.
Manual testing allows for human observation, which may be more beneficial if the goal is enhanced user-friendliness or customer satisfaction.	Automated testing excludes human observation and cannot ensure user-friendliness or a favourable consumer experience.

19. What are pragma directives?

This is a special-purpose directive that is used to enable or disable certain functionality. This type of directive is compiler-specific, meaning it differs from one compiler to the next. The following are some of the #pragma directives:

- **#pragma startup and #pragma exit:** These directives help us describe the routines that must run before the program starts (before control transfers to main()) and right before programme exit (just before control returns from main()).
- **#pragma warn Directive:** This directive is used to hide warning messages that appear during the compilation process. When we have a huge programme and want to solve all of the errors before looking at warnings, we may use this to hide all warnings and focus on the faults. By making little syntax adjustments, we may make the warnings visible once more.
- **#pragma GCC poison:** The GCC compiler supports this directive, which is used to totally eliminate an identifier from a programme. The #pragma GCC poison directive can be used to prevent an identifier from being used.
- **#pragma GCC dependence:** You can use the #pragma GCC dependency to compare the relative dates of two files. A warning is issued if the other file is more recent than the current file. If the current file is derived from the other file and needs to be regenerated, this is handy.
- **#pragma GCC system_header:** This pragma does not accept any arguments. The rest of the code in the current file is processed as though it originated from a system header as a result of this.
- **#pragma once:** The #pragma once directive has a fairly straightforward concept. Even if the programmer includes it numerous times during a compilation, the header file containing this directive is only included once. There is no mention of this in any ISO C++ standard. The #include guard idiom is comparable to this directive. The use of #pragma only once prevents the programme from being optimised for numerous inclusions.

20. How will you delete a node in a doubly-linked list (DLL)?

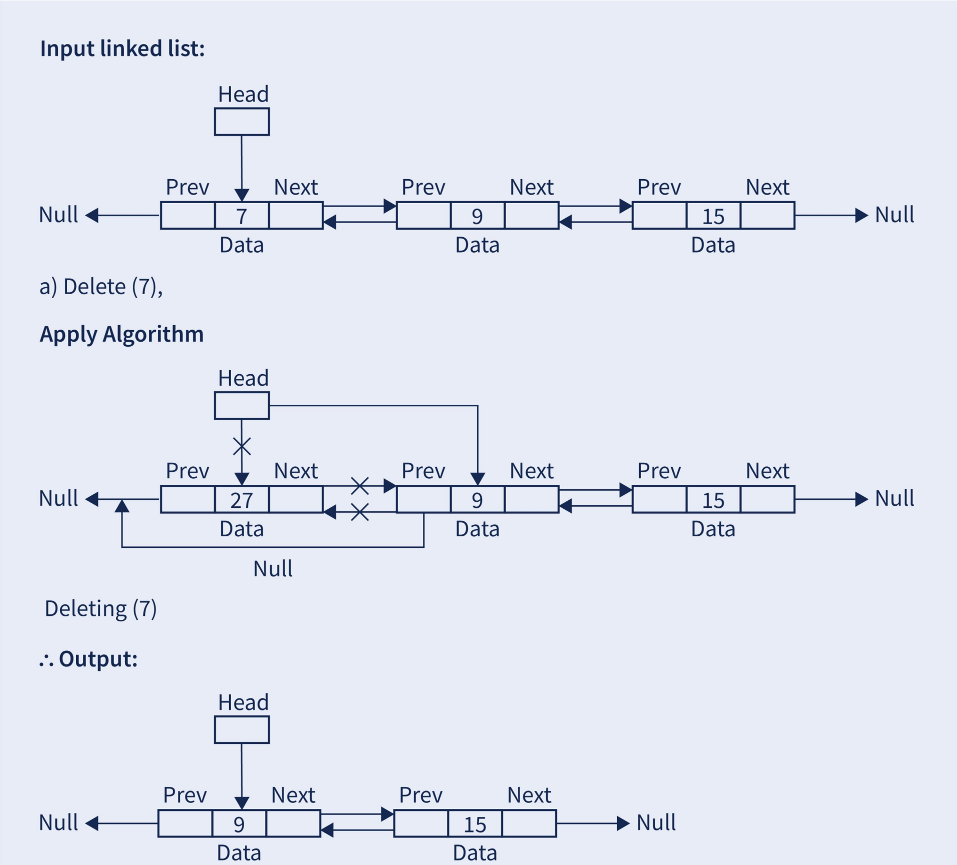
The following is the process to delete a node in a doubly-linked list (DLL):

- Step 1: Create a function that takes as arguments a linked list and a node that needs to be deleted, and deletes the node.

-
- Step 2: To delete a head node:
-
    - Move the head pointer to the next node in the current node (head here).
    - Set the previous pointer of the next node to the previous pointer of the current node.
- Step 3: If you wish to remove the centre node, follow these steps.
-
    - Set the previous pointer of the next node to the previous pointer of the current node.
    - Move the previous node's next pointer to the current node's next pointer.
    - Remove the node. (the most recent node)
    - You do not need to delete previous or next if they are NULL. (for removing the final node)
- Step 4: Call the function on the given linked list and specify the node you want to delete.

Code to delete a node in a doubly-linked list:

```
import gc

# Node of the doubly linked list
class Node:

    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None

class DoublyLinkedList:

    def __init__(self):
        self.head = None

   # A function to delete a node in a Doubly Linked List.

    def deleteNode(self, todelete):

        if self.head is None or todelete is None:
            return

        # If node to be deleted is the head node
        if self.head == todelete:
            self.head = todelete.next

        # Change next only if node to be deleted is NOT
        # the last node
        if todelete.next is not None:
            todelete.next.prev = todelete.prev

        # Change prev only if node to be deleted is NOT
        # the first node
        if todelete.prev is not None:
            todelete.prev.next = todelete.next
        gc.collect()


    # Given a reference to the head of a list and an
    # integer, inserts a new node on the front of list
    def push(self, new_data):

        # Allocates node and inserts the data in it
        new_node = Node(new_data)

        # Make next of new node as head and previous as None
        new_node.next = self.head

        # Change prev of head node to new_node
        if self.head is not None:
            self.head.prev = new_node

        # Move the head to point to the new node
        self.head = new_node


    def printList(self, node):
        while(node is not None):
            print(node.data,end=' ')
            node = node.next

# Start with an empty list
dll = DoublyLinkedList()

dll.push(2);
dll.push(4);
dll.push(8);
dll.push(10);

print ("\n The original linked list",end=' ')
dll.printList(dll.head)

# delete nodes from doubly linked list
dll.deleteNode(dll.head)
dll.deleteNode(dll.head.next)
dll.deleteNode(dll.head.next)
# Updated linked list will be NULL<-8->NULL
print("\n The updated linked list",end=' ')
dll.printList(dll.head)
```



Siemens Interview Preparation

1. Interview Preparation Tips

- If you are a new candidate pursuing a technical position and have a strong academic record, you will have an immediate advantage over other candidates in being considered for Siemens. The ideal method to show yourself in the Siemens interview questions round is to have a firm grasp of your technical knowledge, programming capabilities, and coding talents.
- If you are fluent in the C/C++ programming languages, you will almost certainly be asked to write codes as part of your coding round. They will ask you questions about your CV, make sure to just include those points in your resume for which you are completely prepared and qualified.
- You may also be questioned about your most recent projects, whether by your professors or by your prior employer. It's critical to stay focused on the fundamentals of core areas relevant to your speciality and to improve both your practical and theoretical understanding.
- Take a close look at Siemens' focus areas to see if your interests fit with the goods that they develop. Apply for a position with Siemens if you are interested in working in any of these areas and have the necessary skillset and potential for a successful

career. For the latest employment updates from Siemens, you may either go to their career page on their official website or register with online job portals.

Frequently Asked Questions

1. Is Siemens interview hard?

The initial stages of the Siemens interview include the Online Aptitude Test and Online Technical Test. The difficulty level for this is moderate. Candidates who pass the Aptitude and Technical test at the end of the process are invited to an interview.

2. Why Siemens?

For hiring, they look for people who aren't simply passionate about disruptive ideas but also have the confidence to put them into action.

3. How to get an internship at Siemens?

Students that are interested in applying for an internship at Siemens can do so using the following methods:

- **Online application:** You can first register on the company's website before searching for and applying for open internship positions.
- **Campus recruitment:** Siemens recruits interns at a number of colleges across India. Contact your college's placement coordinator to schedule a campus visit.
- **Emailing the HR executive:** Locate a former/current Siemens employee/intern and request her email address. Directly address your letter to HR, explicitly expressing your purpose and including a great cover letter and résumé.
- **Through a referral:** If you know somebody who works at Siemens, ask her for a recommendation. For this, you can also use Facebook or LinkedIn to find some personnel.

4. How would you describe yourself?

While the open-ended nature of this question may make it seem more intimidating to respond, it actually works in your favour. Because the question is open-ended, you can go in any direction with your response.

Take some time before your interview to consider your primary strengths. What distinguishes you from others in your previous positions? If you've always been good at remaining organized, for example, you could want to include that in your response.

5. How to apply for a job at Siemens?

- The most convenient method is to use the online mode. You build your profile on the Siemens home page. Of course, you'll be given a unique account that provides you access to all of Siemens' various opportunities around the world.
- They have a referral program where you can be recommended by an internal employee.
- On-campus recruitment is also used by Siemens to hire freshmen. It does, however, only visit a few colleges and universities.