INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING (CYBER SEURITY)

DEFINITION AND TERMINOLGY

Course Title	**NETWORK PROGRAMMING AND MANAGMENT**				
Course Code	ACCC05				
Program	B.Tech				
Semester	V	CSE(CS)			
Course Type	Elective				
Regulation	IARE - UG20				
Course Structure	Theory			Practical	
	Lecture	Tutorials	Credits	Laboratory	Credits
	3	-	3	-	-
Course Coordinator	Dr. R Obulakonda Reddy, Professor				

COURSE OBJECTIVES:

The students will try to learn:

I	The basic concepts of connection oriented communication over network.
II	The concepts of multiplexing in client server environment
III	The functions and protocols needed for connection less communication over networks
IV	The management concepts and practical issues of simple network management protocols.

COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO 1	**Interpret** TCP Socket functions between client and server to listen to the TCP port for incoming connections	Understand
CO 2	**Make use of** different boundary conditions in the server and I/O multiplexing to establish the connection in the network	Apply
CO 3	**Match** each of the socket options for each of the layer in the TCP/IP stack to improve the performance of wired network connections	Remember

CO 4	**Recall** the UDP socket functions to maintain low – latency and loss – tolerance connections between applications on the internet	Remember
CO 5	**Demonstrate** the working of different communication protocols that helps to create secure socket applications.	Understand
CO 6	**Illustrate** various network management protocols for monitoring and control of networks on Local Area Network or Wide Area Network.	Understand

DEFINITION AND TERMINOLOGY:

S.No	DEFINITION	CO's
	MODULE I	
	ELEMENTARY TCP SOCKETS	
1	**How Does web browser communicating with a web server?**	CO 1
	Web browsers communicate with web servers using the HyperText Transfer Protocol (HTTP). When you click a link on a web page, submit a form, or run a search, the browser sends an HTTP Request to the server.	
2	**Define Sockets?**	CO 2
	A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.	
3	**What is socket and how it works?**	CO 2
	Sockets are commonly used for client and server interaction. Typical system configuration places the server on one machine, with the clients on other machines. The clients connect to the server, exchange information, and then disconnect. A socket has a typical flow of events.	
4	**What are the types of sockets?**	CO 2
	Socket Types Stream sockets enable processes to communicate using TCP. A stream socket provides a bidirectional, reliable, sequenced, and unduplicated flow of data with no record boundaries. ... Datagram sockets enable processes to use UDP to communicate. ... Raw sockets provide access to ICMP.	
5	**What is the difference between a plug and a socket? .**	CO 2
	A plug is the movable connector attached to an electrically operated device, and the socket is fixed on equipment or a building structure and connected to an energised electrical circuit. The plug is a male connector, often with protruding pins that match the openings and female contacts in a socket	

6	**Define OSI Model** OSI model is a generic model that is based upon functionalities of each layer. TCP/IP model is a protocol-oriented standard. OSI model distinguishes the three concepts, namely, services, interfaces, and protocols. TCP/IP does not have a clear distinction between these three.	CO 1
7	**Difference Between TCP/IP and OSI Model** The TCP/IP or the Transmission Control Protocol/ Internet Protocol is a communication protocols suite using which network devices can be connected to the Internet. On the other hand, the Open Systems Interconnection or OSI Model is a conceptual framework, using which the functioning of a network can be described.	CO 2
8	**What do you mean by TCP IP suite?** Transmission Control Protocol/Internet Protocol TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP is also used as a communications protocol in a private computer network (an intranet or extranet).	CO 2
9	**Why is TCP IP called a suite?** The name "TCP/IP" refers to an entire suite of data communications protocols. The suite gets its name from two of the protocols that belong to it: the Transmission Control Protocol (TCP) and the Internet Protocol (IP).	CO 2
10	**What are the 4 core protocols in the TCP IP suite?** The Internet Protocol (IP) The Internet Control Message Protocol (ICMP) The Internet Group Management Protocol (IGMP) The Address Resolution Protocol (ARP)	CO 2
11	**What are the 5 layers of the TCP IP suite?** The TCP/IP model is based on a five-layer model for networking. From bottom (the link) to top (the user application), these are the physical, data link, net- work, transport, and application layers.	CO 1
12	**What is 3 way TCP handshake.** The TCP handshake TCP uses a three-way handshake to establish a reliable connection. The connection is full duplex, and both sides synchronize (SYN) and acknowledge (ACK) each other. The exchange of these four flags is performed in three steps—SYN, SYN-ACK, and ACK	CO 1
13	**What is SYN and SYN-ACK?** Known as the "SYN, SYN-ACK, ACK handshake," computer A transmits a SYNchronize packet to computer B, which sends back a SYNchronize-ACKnowledge packet to A. Computer A then transmits an ACKnowledge packet to B, and the connection is established.	CO 1

14	**What happens if TCP SYN is dropped? Packet drops**	CO 11
	This scenario denotes that the network device between the source and destination is dropping the packets. If the initial TCP handshake is failing because of packet drops, then you would see that the TCP SYN packet is retransmitted only three times..	
15	**Define socket data structure**	CO 11
	The socket data structure defines the socket. During a socket subroutine, the system dynamically creates the socket data structure. The socket address is specified by a data structure that is defined in a header file.	
16	**What is the purpose of a socket address? Socket addresses**	CO
	An application can communicate with a remote process by exchanging data with TCP/IP by knowing the combination of protocol type, IP address, and port number. This combination is often known as a socket address. It is the network-facing access handle to the network socket.	
17	**How socket address is created?**	CO 11
	TCP/IP creates the socket address as an identifier that is unique throughout all Internet networks. TCP/IP concatenates the Internet address of the local host interface with the port number to devise the Internet socket address. With TCP/IP, sockets are not tied to a destination address.	
18	**How socket address structure passes from process to kernal.**	CO 11
	The four functions accept(), recvmsg(), getsockname() and getpeername() pass a socket address structure from kernal to the process, the reverse direction form the precious scenario. In this case the length is passed as pointer to an integer containing the size of structure.	
19	**Define ICMPv6?**	CO 11
	ICMPv6 has a framework for extensions to implement new features. Several extensions have been published, defining new ICMPv6 message types as well as new options for existing ICMPv6 message types.	
MODULE II		
APPLICATION DEVELOPMENT		
1	**ow to implement TCP sockets in ?**	CO 3
	TCP sockets are used for communication between a server and a client process. The server's code runs first, which opens a port and listens for incoming connection requests from clients. Once a client connects to the same (server) port, the client or server may send a message.	

2	**Define TCP Echo Server ?** TCP Echo Server In the TCP Echo server , we create a socket and bind to a advertized port number. After binding , the process listens for incoming connections. Then an infinite loop is started to process the client requests for connections.	CO 3
3	**What is TCP echo client?** In the TCP Echo client a socket is created. Using the socket a connection is made to the server using the connect() function. After a connection is established , we send messages input from the user and display the data received from the server using send() and read() functions.	CO 5
4	**What is POSIX signal handling?** A signal is a notification to a process that an event has occurred. Signals are sometimes called software interrupts. Signals usually occur asynchronously. By this we mean that a process doesn't know ahead of time exactly when a signal will occur.	CO 5
5	**What is signal handling in network programming?** Signal, or a software interrupt, is a notification to a process that an event has occurred. Generally, signals are asynchronous, because a process doesn't know that or when a signal will occur. Signals are sent to a process by - another process, or itself, or the kernel.	CO 3
6	**What happens after signal handling?** After processing a signal, you may want the program to continue execution from the point at which it was interrupted. In this case, the handler simply executes a return statement. However, for some signals (such as SIGILL), it is impossible to resume at the point at which the signal occurred.	CO 7
7	**Define signal function in networking** An easier way to set the disposition for a signal is to call the signal function. The first argument is the signal name and the second arguments is the pointer to a function or one of the constants SIG_IGN or SIG_DFE. Normally, the define our own signal function that just calls the Posix sigaction	CO 3
8	**Define Interrupted System Calls** Interruption of a system call by a signal handler occurs only in the case of various blocking system calls, and happens when the system call is interrupted by a signal handler that was explicitly established by the programmer	CO 5
9	**Can a system call interrupt another system call?** Yes. The difference being that after returning from the interrupt, a reschedule may happen immediately, without waiting for the interrupted system call to finish.	CO 5

10	**What is a SIGPIPE signal?** A SIGPIPE is sent to a process if it tried to write to a socket that had been shutdown for writing or isn't connected (anymore). To avoid that the program ends in this case, you could either. make the process ignore SIGPIPE	CO 5
11	**Define poll function** The poll() method attempts to consolidate the arguments of select() and provides notification of a wider range of events. The SUSv2 defines poll() as follows: int poll(struct pollfd fds, nfds_t nfds, int timeout)	CO 6
12	**Define I/O multiplexing** I/O multiplexing is the capability to tell the kernel that we want to be notified if one or more I/O conditions are ready, like input is ready to be read, or descriptor is capable of taking more output	CO 6
13	**What is signal driven I/O?** Historically, this has been called asynchronous I/O, but the signal-driven I/O that we will describe is not true asynchronous I/O. The latter is normally defined as the process performing the I/O operation (say a read or write), with the kernel returning immediately after the kernel initiates the I/O operation.	CO 1
14	**What is blocking IO model?** With blocking I/O, when a client makes a request to connect with the server, the thread that handles that connection is blocked until there is some data to read, or the data is fully written. Until the relevant operation is complete that thread can do nothing else but wait.	CO 7
15	**What are different IO models?** Synchronous I/O versus Asynchronous I/O Using these definitions, the first four I/O models—blocking, nonblocking, I/O multiplexing, and signal-driven I/O—are all synchronous because the actual I/O operation (recvfrom) blocks the process. Only the asynchronous I/O model matches the asynchronous I/O definition.	CO 5
16	**Define Select Function** The Select function is used to select between TCP and UDP sockets. This function gives instructions to the kernel to wait for any of the multiple events to occur and awakens the process only after one or more events occur or a specified time passes.	CO 5
17	**What is select function in socket programming?** select() The select() call monitors activity on a set of sockets looking for sockets ready for reading, writing, or with an exception condition pending.	CO 7

18	**Define The shutdown() function** The shutdown() function shall cause all or part of a full-duplex connection on the socket associated with the file descriptor socket to be shut down. Specifies the file descriptor of the socket. Specifies the type of shutdown.	CO 5
19	**Do sockets close?** The gum tissue should close off the extraction site within a matter of days. Within about two weeks, there should be a smooth texture over the socket that matches the gingiva (gum tissues) surrounding it. Underneath the gingiva, however, it may be around a few months before the socket starts to close	CO 3
20	**Synchronous vs asynchronous**	CO 3
	A synchronous I/O operation causes the requesting process to be blocked until that I?O operation is completes. An asynchronous I/O operation does not cause the requesting process to be blocked. Based on this definition, the first four are synchronous as the actual I/O operation blocks the process. Only asynchronous I/O model matches the asynchronous I/O definition.	
MODULE III		
SOCKET OPTIONS, ELEMENTARY UDP SOCKETS		
1	**Define socket option in network programming.** In addition to binding a socket to a local address or connecting it to a destination address, application programs need a method to control the socket. For example, when using protocols that use time out and retransmission, the application program may want to obtain or set the time-out parameters.	CO 3
2	**Define generic socket option in network programming.** These socket options are protocol independent meaning that the protocol independent code within the kernel handles these and not particular module of any protocol. Some options apply to only certain types of sockets.	CO 5
3	**Deine SO_BROADCAST** This socket option enables or disables the ability of the process to send broadcast messages. Broadcasting is supported for only datagram sockets and only on net works such as ethernet, token ring etc but not point to point networks. This option controls whether datagrams may be broadcast from the socket. The value has type int; a nonzero value means “yes”.	CO 3
4	**Define SO_DEBUG?** The option is supported by TCP only. When enabled for a TCP socket, the kernal keeps track of all the packets sent or received by TCP for the socket.	CO 6

5	**Define SO_DONTROUTE Socket option**	CO 5
	The option specifies that outgoing packets are to bypass the normal routing mechanism of the underlying ptotocol. With Ipv4, the packet is directed to the appropriate local interface, as specifried by the network and subnet protions of the destination address. If the local interface cannnot be determined from destination address, ENETUNREACH is returned.	
6	**Define SO_ERROR options.**	CO 5
	RThis option can be used with getsockopt only. It is used to reset the error status of the socket. When a errors occurs on a socket, the protocol module sets a variable named so_error for that socket to one of the standard unix values. The processs is immediately notified. The process can then obtain the values of so-error by fetching the SO_ERROR socket option. After the receipt, the so_error value is reset to 0 by the kernel.	
7	**Define SO_KEEPALIVE socket option?**	CO 3
	When the keep alsive socket option is set for a TCP socket, and if no data is exchanged across in either direction for two hours TCP automatically sends a keepalive probe to the peer. The probe is a TCP segment to which the peer must respond.	
8	**Define application ACK.**	CO 5
	An application acknowledgement letter is a business letter that employers may send to job candidates to inform them of the status of their job application. It's helpful for companies to acknowledge when they receive resumes and are reviewing a candidate's application	
9	**Define SOOOBINLINE socket option**	CO 7
	SO_OOBINLINE. When the OOBINLINE option is set, any TCP urgent data received on the socket will be received through the socket input stream. static int. SO_RCVBUF. Set a hint the size of the underlying buffers used by the platform for incoming network I/O.	
10	**Define SO-OOBINLINE socket option?**	CO 3
	SO_OOBINLINE. When the OOBINLINE option is set, any TCP urgent data received on the socket will be received through the socket input stream. static int. SO_RCVBUF. Set a hint the size of the underlying buffers used by the platform for incoming network IO.	
11	**Define SO_KEEPALIVE?**	CO 3
	This option controls whether the underlying protocol should periodically transmit messages on a connected socket. If the peer fails to respond to these messages, the connection is considered broken. The value has type int; a nonzero value means "yes".	
12	**Define SO_LINGER**	CO 3
	This option specifies what should happen when the socket of a type that promises reliable delivery still has untransmitted messages when it is closed; see Closing a Socket. The value has type struct linger	

13	**Define SO_BROADCAST** This option controls whether datagrams may be broadcast from the socket. The value has type int; a nonzero value means "yes"	CO 4
14	**Define SO_SNDBUF.** This option gets or sets the size of the output buffer. The value is a size_t, which is the size in bytes.	CO 4
15	**Define SO_RCVBUF and SO_SNDBUF Socket Option** The receive buffer are used by the TCP and UDP to hold received data until it is read by the application. With TCP, the available room in the socket receive buffer is the window that TCP avertises to the other end. Hence the the peer sends only that amount of data and any data beyond that limit is discarded. In case of UDP, the bufffer size is not advertised hence, any data that do not fit ito the buffer, are dropped However, the abovementioned socket options allows one to change the default sizes. The default values for the TCP and UDP differ for different implementation. It is normally 4096 for TCP and send buffer for UDP is 9000 and 40000 bytes for receive buffer.	CO 3
16	**Define SO_REUSEADDR and SO_REUSEPORT: .** The SO_REUSEADDR serves four purposes : This option allows a listening server to start and bind its well known port even if previously established connections exists that use this port as their local port. As the server is in listening state, when connection request comes from a client, a child process is spawned to handle that client. Wqith this listening server terminates. Once again when the listening is restarted buy calling socket, bind and listen, the call to bind fails because the listening server is trying to bind a port that is part of exisitng connection. But if the server is sets the SO_REUSEADDR socket option between the calls to socket and bind, the latter function will succeed.	CO 7
17	**Define SO_TYPE socket Option: ?** SO_TYPE socket Option: This option returs the socket ytpe. The integer value returned is a value such as SOCK _ STREAM or SOCK_DGRAM.	CO 3
18	**Define SO_USELOOPBACK Socket Option.** When this option is set, the socket receives a copy of everything send on the socket.	CO 6
19	**Define IP_HDRINC Socket Option** If this socket is set for a raw socket, we must buid our own IP header for all datagrams that we send on the raw socket. Normally kernel builds the headers for datagrams sent on raw socket. But for some applications, build there own IP address to override that IP would place into certain header fields. (Traceroute).	CO 3

20	**Deine IP_OPTIONS Socket Options**	CO 6
	Setting this option allows us to set the IP option in the Ipv4 header. This requires intimate knlowledge of the format of the IP options in the IP header.	
	MODULE IV	
	ADVANCED SOCKETS	
1	**Define threats in network programming.**	CO 9
	A thread is a separate computational process that can run in parallel with other threads. When a program uses threads to do network communication, it is possible that some threads will be blocked, waiting for incoming messages, but other threads will still be able to continue performing useful work.	
2	**What is thread and process?**	CO 9
	A process, in the simplest terms, is an executing program. One or more threads run in the context of the process. A thread is the basic unit to which the operating system allocates processor time. A thread can execute any part of the process code, including parts currently being executed by another thread.	
3	**Why do we need threads?**	CO 9
	Thread is a light weight process which helps in running the tasks in parallel. The threads works independently and provides the maximum utilization of the CPU, thus enhancing the CPU performance. Threads to make Java application faster by doing multiple things at same time.	
4	**Define the thread functions?**	CO 9
	Thread functions allow users to implement concurrent functions at the same time, which can either be dependent on each other for execution or independent	
5	**Define pthread_atttr_t**	CO 8
	Each thread has a number of attributes – priority, initi8al stack size, whether is demon thread or not. If this variable is specified, it overrides the default. To accept the default, attr argument is set to null pointer.	
6	**Define func?**	CO 8
	When the thread is created, a function is specified for it to execute. The thread starts by calling this function and then terminates either explicitly (by calling pthread_exit) or implicitly by letting this function to return. The address of the function is specified as the func argument. And this function is called with a single pointer argument, arg. If multiple arguments are to be passed, the address of the structure can be passed	

7	**Define pthread_join function**	CO 8
	int pthread_join (pthread_t tid, void status) We can wait for a given thread to terminate by calling pthread_join . We must specify the tid of the thread that we wish to wait for. If the status pointer is non null, the return value from the thread is stored in the location pointed to by status.	
8	**Define pthread_self function**	CO 9
	Each thread has an ID that identifies it within a given process. The thread ID is returned by pthread_create. This function fetches this value for itself by using this function: pthread_t pthread_self(void).	
9	**Define pthread_detach function.**	CO 2
	A thread is joinable (the default) or detached. When a joinable thread terminates, its thread ID and exit status are retained until thread calls pthread_join(). But a detached thread for example daemon thread- when it terminates all its resources are released and we cannot wait for it terminate. When one thread needs to know when another thread terminates, it is best to leave the thread joinable. Int pthread_detach (pthread_t tid);	
10	**Define pthread_exit function**	CO 8
	One way for the thread to terminate is to call pthread_exit(). void pthread_exit (void status);	
11	**Define raw sockets.**	CO 5
	A raw socket is a type of socket that allows access to the underlying transport provider. This topic focuses only on raw sockets and the IPv4 and IPv6 protocols. This is because most other protocols with the exception of ATM do not support raw sockets.	
12	**What is a raw socket in Linux?**	CO 8
	RAW-sockets are an additional type of Internet socket available in addition to the well known DATAGRAM- and STREAM-sockets. They do allow the user to see and manipulate the information used for transmitting the data instead of hiding these details, like it is the case with the usually used STREAM- or DATAGRAM sockets.	
13	**Define Internet Control Message Protocol (ICMP).**	CO 2
	Internet Control Message Protocol (ICMP) ICMP is used for status reporting over the Internet. ICMP provides error and control messages that are used in a variety of situations. For example, when an IP packet cannot be delivered due to a destination not being reachable a Destination Unreachable ICMP message is used.	
14	**Define RAW SOCKET CREATION**	CO 19
	To create a socket of type SOCK_RAW, call the socket or WSASocket function with the af parameter (address family) set to AF_INET or AF_INET6, the type parameter set to SOCK_RAW, and the protocol parameter set to the protocol number required.	

15	**What is the purpose of raw socket?** The raw socket interface provides direct access to lower layer protocols, such as the Internet Protocol (IP) and Internet Control Message Protocol (ICMP or ICMPv6). You can use raw sockets to test new protocol implementations.	CO 9
16	**Define Raw Socket Output.** The output of raw socket is governed by the following rules 1. Normal output is performed by calling send to or sendmsg and specifying the destination IP address. IN case the socket has been connected, write and send functions can be used. 2. If the IP_HDRINCL option is not set, the IP header will be built by the kernal and it will be prepend it to the data. 3. If IP_HDRINCL is set, the header format will remain the same and the process builds the entire IP header except the IPv4 identification field which is set to 0 by the kernel 4. The kernel fragments the raw packets that exceed the outgoing interface.	CO 5
17	**What is ping program explain?** A ping (Packet Internet or Inter-Network Groper) is a basic Internet program that allows a user to test and verify if a particular destination IP address exists and can accept requests in computer network administration. The acronym was contrived to match the submariners' term for the sound of a returned sonar pulse.	CO 9
18	**What is meant by mutexs ?** In computer programming, a mutex (mutual exclusion object) is a program object that is created so that multiple program thread can take turns sharing the same resource, such as access to a file.	CO 9
19	**Is ICMP and ICMPv6 the same?** ICMP is available for both IPv4 and IPv6. ICMPv4 is the messaging protocol for IPv4. ICMPv6 provides these same services for IPv6 but includes additional functionality. In this course, the term ICMP will be used when referring to both ICMPv4 and ICMPv6.	CO 9
20	**What is the ICMP protocol?** The Internet Control Message Protocol (ICMP) is a protocol that devices within a network use to communicate problems with data transmission. In this ICMP definition, one of the primary ways in which ICMP is used is to determine if data is getting to its destination and at the right time.	CO 9
MODULE V		
SIMPLE NETWORK MANAGEMENT		
1	**Define Simple Network Management Protocol.** Simple Network Management Protocol (SNMP) is an application-layer protocol for monitoring and managing network devices on a local area network (LAN) or wide area network (WAN).	CO 10

2	**Define SNMP components?** An SNMP system consists of four key components: network management system (NMS), SNMP agent, managed object, and management information base (MIB). The NMS manages network elements on a network. Each managed device contains an SNMP agent process, a MIB, and multiple managed objects.	CO 6
3	**What are the different types of SNMP?** There are three types of SNMP traps: standard, built-in, and user-defined. A trap can be used to periodically check for different operational thresholds or failures, which are defined in the MIB.	CO 6
4	**What is management information base in SNMP?** A management information base (MIB) is a Simple Network Management Protocol (SNMP) flat-file, nonrelational database that describes devices being monitored. Network management platforms monitor nodes by reading the value of the managed resources in the MIB.	CO 6
5	**What is the function of the SNMP manager?** SNMP Manager: An SNMP Manager is an application that performs the operational roles of generating requests to modify and retrieve management information and receiving the requested information and trap-event reports that are generated by the SNMP agent.	CO 6
6	**How does SNMP define MIB?** An SNMP-compliant MIB contains definitions and information about the properties of managed resources and the services that the agents support. The manageable features of resources, as defined in an SNMP-compliant MIB, are called managed objects or management variables (or just objects or variables).	CO 6
7	**What is MIB structure?** The structure of Management Information Base (MIB) is a formatted text file that lists all of the data objects used by a particular piece of equipment. When you buy a monitor device that uses SNMP (for example, a managed switch), you'll tell it to send messages to your central SNMP manager.	CO 6
8	**What is standard MIB?** The MIBs for a specific company's device are called an enterprise MIB. Standards have been made describing the format of common data. These are called standard MIBs. A special set of these MIBs are called MIB-II. These are MIBs for data you would expect every network device to have such as Ethernet, TCP, and UDP.	CO 12
9	**List out Network Management Requirements** Fault Management, Performance Management, Account Management, Configuration and Name Management, Security management,	CO 6

10	**Define Network Management Configuration?** The architecture of the network management system is shown in the next page. Each node contains a collection of software devoted to the network management tasks referred as NME – Network Management Entity.	CO 10
11	**Define Network Management Software Architecture.** Network Management Software Architecture can be divided into three broad categories; 1. User Presentation software 2. Network Management Software 3. Lowest level of management specific software	CO 6
12	**Define proxy.** A proxy server is a system or router that provides a gateway between users and the internet. Therefore, it helps prevent cyber attackers from entering a private network. It is a server, referred to as an "intermediary" because it goes between end-users and the web pages they visit online.	CO 6
13	**Define MIB** A management information base (MIB) is a Simple Network Management Protocol (SNMP) flat-file, nonrelational database that describes devices being monitored. Network management platforms monitor nodes by reading the value of the managed resources in the MIB.	CO 12
14	**What is Structure Management Information.** Structured Management Information explains "How to write and define MIB". The SMI defines the general framework within which a MIB can be defined and constructed . The SMI identifies the data type that can be used in the MIB and specifies how within MIB are represented and named	CO 6
15	**Write a short note on MIB structure.** MIB Structure The Internet Naming Hierarchy Objects Types Simple/Tabular Objects Instances Identification	CO 6

Course Coordinator: **HOD CSE(CS)**
Dr. R Obulakonda Reddy Professor