00980

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 3, Semester 1 (2022)

IT3010 – Network Design and Management

Duration: 2 Hours

June 2022

Instructions to Candidates:

- This paper has 4 questions.
- Answer all questions in the booklet given.
- The total marks for the paper is 100.
- This paper contains 12 pages, including the cover page.
- Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.
- This is not an open book examination.

Question 1 [25 Marks]

Read the paragraph given below and answer the question.
QwiCom is a middle scale software development company which provide software-based solutions for various companies, individuals both as developers and service providers. The company is going through a major network upgrade where a new domain name *qwicom.co* is attached with 3 new branches. There are multiple number of running servers such as web servers, email servers, file servers, application servers etc., as well as number of stand-alone and connected host machines which needs upgrades to the operating system and its configurations for a smooth network operational purpose. A new auditing mechanism is to be implemented since now there will be remote working connections between the main and the branch offices via VPN technology. Furthermore, the network data/information exchanged needs to be secured and the new domain admins will play a major role in implementing permission levels for login and resource sharing purposes. a new monitoring tool should be introduced to monitor the network activities in main office as well as within the branches which would help in auditing and accounting.

a. Named 5 categories in ISO management framework. (5 marks)

b. Interpret the management categories that should be covered bye QwiCom company for their major network upgrade.
(4 marks)

c. Analyze and provide 2 improvement area for each management category identified in Part b. (8 marks)

d. Discover three focus areas that should be given attention aside from the given description. (3 marks)

e. Criticize how performance management and configuration management we'll go hand in hand with example taken from the paragraph given above. (5 marks)

Question 2 [25 Marks]

a. What are the purpose of the version and of the community components of SMNP?

(2 marks)

b. The SNMP definition provides an event detection mechanism. List 4 SNMP events that can be detected and state how a manufacture of any device can signal that one of their events has occurred. (4 marks)

c. Note the abbreviated MIB II Definition from RFC 1213 on Appendix II. Assume you are to using SNMP to determine the route of a packet from a source to a destination.

i. One of the steps is to look up the routing table to find the next hop. What is the Object Identifier for the next hop if the destination is 134.7.0.0? (4 marks)

ii. To get the OID mentioned in part I, as the next entry using *snmpgetnext*, what current Object Identifier should be given? (2 marks)

iii. Name a field in the routing table that can be written (and so altered if there is a fault)? (2 marks)

iv. Name an object in the MIB that can be monitored to detect errors indicating that the routing table might be faulty? (1 mark)

d. Assume the following is that part of an SNMP message corresponding to a VarBindList.

Note to decode a SNMP packet, you need the ASN.1 definition of SNMP (Appendix I), a
MIB (Appendix II) and the BER encoding rules and ASCII codes (Appendix III).

Further, the MIB contains objects with the syntax Counter and the syntax IpAddress. Note, in order to compute the tag, Counter is and an Integer, while IpAddress is an OCTET STRING of length 4.

The following is a VarBindList (that is, part of the SNMP message) that has been captured.

Decode it.
30 0F 30 0D 06 08 2B 06 01 02 01 04 05 00 41 01 0A

(10 marks)

Question 3 [25 Marks]

a. Consider the following diagram figure 1. Assume all the servers are nodes on the one server.

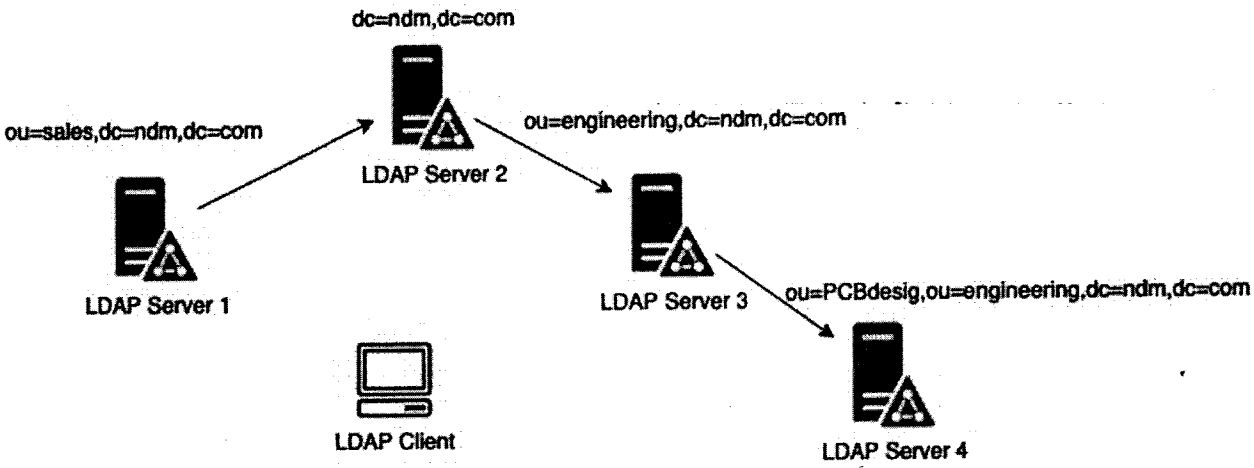

Figure 1 – Active Directory Tree

Generate an LDIF file to create all the LDAP Server nodes. (8 marks)

b. Concerning LDAP:

i. Why are hierarchical structures useful in network management? (5 marks)
ii. What is the difference between a DIT and an Object Identifier Tree? (3 marks)
iii. Define a LDAP Schema? (3 marks)
iv. List the components in a LDAP Entry? (3 marks)
v. Briefly describe the definition of an LDAP Attribute? (3 marks)

Question 4 [25 Marks]

a. List down basic network server types and briefly explain (4 Marks)

b. A student who has logged into the *sliit.lk* domain is trying to request a web page www.itmasters.edu via chrome web browser. The domain *sliit.lk* has its own local domain name server. Show the steps for resolving web page www.itmasters.edu using a sketch (8 Marks)

c. Briefly explain a DNS zone? (3 Marks)

d. Analyze the given forward lookup zone file for the DNS implementation and answer the following questions? (10 Marks)

```
$TTL 86400
@       IN      SOA     dekin_dc.edu.au     root.edu.au     (
        2011071001          ;Serial
        3600                ;Refresh
        1800                ; Retry
        604800              ; Expire
        86400       )       ;Minimum TTL

@       IN      NS      dekin_dc.edu.au
@       IN      A       192.168.0.10
@       IN      A       192.168.0.50
dekin_dc            IN      A       192.168.0.10

dekin_cl            IN      A       192.168.0.50
```

i. What is the server ip address?
ii. What is the client ip address?
iii. What is the server host name?
iv. What is the domain name?
v. By analyzing the above forward lookup zone, complete the given reverse look up zone file below?

```
$TTL 86400
@       IN      SOA     dekin_dc.edu.au     root.edu.au     (
        2011071001          ;Serial
        3600                ;Refresh
        1800                ; Retry
        604800              ; Expire
        86400       )       ;Minimum TTL

@ IN NS     _____
@ IN PTR    _____
server IN A _____
client IN A _____
__ IN PTR   dekin_dc.edu.au
__ IN PTR   dekin_cl
```

Appendix I

SNMP Definition

```
RFC1157-SNMP DEFINITIONS ::= BEGIN
  IMPORTS
      ObjectName, ObjectSyntax, NetworkAddress, IpAddress, TimeTicks
            FROM RFC1155-SMI;
      -- top-level message
      Message ::= SEQUENCE
            { version INTEGER {version-1(0)}} -- version-1 for this RFC
            , community OCTET STRING -- community name
            , data CHOICE
                  { get-request GetRequest-PDU
                  , get-next-request GetNextRequest-PDU
                  , get-response GetResponse-PDU
                  , set-request SetRequest-PDU
                  , trap Trap-PDU
                  }
            }
      -- PDUs
      GetRequest-PDU ::= [0] IMPLICIT PDU
      GetNextRequest-PDU ::= [1] IMPLICIT PDU
      GetResponse-PDU ::= [2] IMPLICIT PDU
      SetRequest-PDU ::= [3] IMPLICIT PDU
      PDU ::= SEQUENCE
            { request-id INTEGER
            , error-status INTEGER
                  { noError(0), tooBig(1), noSuchName(2)
                  , badValue(3), readOnly(4), genErr(5)
                  } -- sometimes ignored
            , error-index INTEGER -- sometimes ignored
            , variable-bindings VarBindList -- values are sometimes
      ignored
            }
      Trap-PDU ::= [4] IMPLICIT SEQUENCE
            { enterprise OBJECT IDENTIFIER -- type of object generating
      trap
            , agent-addr NetworkAddress -- address of object generating
      trap
            , generic-trap INTEGER
                  { coldStart(0), warmStart(1), linkDown(2)
                  , linkUp(3), authenticationFailure(4)
                  , egpNeighborLoss(5), enterpriseSpecific(6)
                  } -- generic trap type
            , specific-trap INTEGER -- specific code
            , time-stamp TimeTicks -- time elapsed from last net init
            , variable-bindings VarBindList -- "interesting" information
            }
      -- variable bindings
      VarBind ::= SEQUENCE
            { name ObjectName
            , value ObjectSyntax
            }
      VarBindList ::= SEQUENCE OF VarBind
END
```

Appendix I

MIB II Definition

```
        RFC1213-MIB DEFINITIONS ::= BEGIN
        IMPORTS mgmt, NetworkAddress, IpAddress,
Counter, Gauge,
                TimeTicks FROM RFC1155-SMI
                OBJECT-TYPE FROM RFC-1212;
        mib-2 OBJECT IDENTIFIER ::= { mgmt 1 } --
mgmt is 1.3.6.1.2

        DisplayString ::= OCTET STRING
        PhysAddress ::= OCTET STRING

        system OBJECT IDENTIFIER ::= { mib-2 1 }
        interfaces OBJECT IDENTIFIER ::= { mib-2
2 }
        ip OBJECT IDENTIFIER ::= { mib-2 4 }
        icmp OBJECT IDENTIFIER ::= { mib-2 5 }
        tcp OBJECT IDENTIFIER ::= { mib-2 6 }
        udp OBJECT IDENTIFIER ::= { mib-2 7 }
        egp OBJECT IDENTIFIER ::= { mib-2 8 }
        snmp OBJECT IDENTIFIER ::= { mib-2 11 }

                -- the IP group
                -- Implementation of the IP
        group is mandatory for all
                -- systems.

        ipForwarding OBJECT-TYPE
                SYNTAX INTEGER {
                        forwarding(1), -- acting
                as a gateway
                        not-forwarding(2) -- NOT
                acting as a gateway
                        }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                        "The indication of
                        whether this entity is
                        acting
                        as an IP gateway in
                        respect to the
                        forwarding of
                        datagrams received by,
                        but not addressed to,
                        this
                        entity. IP gateways
                        forward datagrams. IP
                        hosts
                        do not (except those
                        source-routed via the
                        host).
                        Note that for some
                        managed nodes, this
                        object may
                        take on only a subset of
                        the values possible.
                        Accordingly, it is
                        appropriate for an agent
                        to
                        return a 'badValue'
                        response if a management
                        station attempts to
                        change this object to an
                        inappropriate value."
                ::= { ip 1 }
        ipDefaultTTL OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                        "The default value
                        inserted into the Time-
                        To-Live
                        field of the IP header
                        of datagrams originated
                        at
                        this entity, whenever a
                        TTL value is not
                        supplied
                        by the transport layer
                        protocol."
                ::= { ip 2 }
        ipInReceives OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The total number of
                        input datagrams received from
                        interfaces, including
                        those received in error."
                ::= { ip 3 }
        ipInHdrErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The number of input
                        datagrams discarded due to
                        errors in their IP
                        headers."
                ::= { ip 4 }
        ipInAddrErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The number of input
                        datagrams discarded because of
                        invalid the IP address
                        in their IP header's
                        destination."
                ::= { ip 5 }
        ipForwDatagrams OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The number of input
                        datagrams for which this
                        entity was not their
                        final IP destination."
                ::= { ip 6 }
        ipInUnknownProtos OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The number of locally-
                        addressed datagrams
                        with unknown or
                        unsupported protocol."
                ::= { ip 7 }
        ipInDiscards OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The number of input IP
                        datagrams discarded
                        for lack of buffer
                        space."
                ::= { ip 8 }
        ipInDelivers OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "The total number of
                        input datagrams
                        successfully
                        delivered to IP user-
                        protocols (including
                        ICMP)."
                ::= { ip 9 }
```

```
ipOutRequests OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The total number of IP
                datagrams which local IP
                user-protocols supplied
                to IP for transmission"
        ::= { ip 10 }
ipOutDiscards OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of output IP
                datagrams discarded
                for lack of buffer
                space."
        ::= { ip 11 }
ipOutNoRoutes OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
                datagrams discarded
                because no
                route could be found to
                transmit."
        ::= { ip 12 }
ipReasmTimeout OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The maximum number of
                seconds which received
                fragments are held while
                they are awaiting
                reassembly."
        ::= { ip 13 }
ipReasmReqds OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
                fragments received which
                needed
                to be reassembled at
                this entity."
        ::= { ip 14 }
ipReasmOKs OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
        datagrams successfully
        reassembled."
        ::= { ip 15 }
ipReasmFails OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of failures
                detected by the IP
                reassembly
                algorithm."
        ::= { ip 16 }
ipFragOKs OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
                datagrams that have been
                successfully fragmented
                at this entity."
        ::= { ip 17 }
ipFragFails OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
                datagrams that have been
                discarded because they
                needed to be fragmented
                but
                their Don't Fragment
                flag was set."
        ::= { ip 18 }
ipFragCreates OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of IP
                datagram fragments that
                have
                been generated as a
                result of fragmentation
                at
                this entity."
        ::= { ip 19 }
        -- the IP address table
        -- The IP address table contains
        this entity's IP addressing
        -- information.
ipAddrTable OBJECT-TYPE
        SYNTAX SEQUENCE OF IpAddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The table of addressing
                information relevant to
                this entity's IP
                addresses."
        ::= { ip 20 }
ipAddrEntry OBJECT-TYPE
        SYNTAX IpAddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The addressing
                information for one of
                this
                entity's IP addresses."
        INDEX { ipAdEntAddr }
        ::= { ipAddrTable 1 }
        IpAddrEntry ::=
        SEQUENCE { ipAdEntAddr
        IpAddress,
        ipAdEntIfIndex INTEGER,
        ipAdEntNetMask IpAddress,
        ipAdEntBcastAddr INTEGER,
        ipAdEntReasmMaxSize INTEGER
        (0..65535)
        }
ipAdEntAddr OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The IP address to which
                this entry's addressing
                information pertains."
        ::= { ipAddrEntry 1 }
ipAdEntIfIndex OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The index value which
                uniquely identifies the
                interface to which this
                entry is applicable."
        ::= { ipAddrEntry 2 }
ipAdEntNetMask OBJECT-TYPE
```

```
        SYNTAX IpAddress
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The subnet mask
                associated with the IP
                address of
                this entry."
        ::= { ipAddrEntry 3 }
ipAdEntBcastAddr OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The value of the least-
                significant bit in the
                IP
                broadcast address used
                for sending datagrams on
                the (logical) interface
                associated with the IP
                address of this entry.
                For example, when the
                Internet standard all-
                ones broadcast address
                is
                used, the value will be
                1. This value applies to
                both the subnet and
                network broadcasts
                addresses
                used by the entity on
                this (logical)
                interface."
        ::= { ipAddrEntry 4 }
ipAdEntReasmMaxSize OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The size of the largest
                IP datagram which this
                entity can re-assemble
                from incoming IP
                datagrams."
        ::= { ipAddrEntry 5 }
        -- the IP routing table
        -- The IP routing table contains
        an entry for each route
        -- presently known to this
        entity.
ipRouteTable OBJECT-TYPE
        SYNTAX SEQUENCE OF IpRouteEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "This entity's IP
        Routing table."
        ::= { ip 21 }
ipRouteEntry OBJECT-TYPE
        SYNTAX IpRouteEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A route to a particular
        destination."
        INDEX { ipRouteDest }
        ::= { ipRouteTable 1 }
IpRouteEntry ::=
        SEQUENCE { ipRouteDest
        IpAddress,
                ipRouteIfIndex INTEGER,
                ipRouteMetric1 INTEGER,
                ipRouteMetric2 INTEGER,
                ipRouteMetric3 INTEGER,
                ipRouteMetric4 INTEGER,
                ipRouteNextHop
                IpAddress,
                ipRouteType INTEGER,
                ipRouteProto INTEGER,
                ipRouteAge INTEGER,
                ipRouteMask IpAddress,
                ipRouteMetric5 INTEGER,
                ipRouteInfo OBJECT
                IDENTIFIER
        }
ipRouteDest OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The destination IP
        address of this route."
        ::= { ipRouteEntry 1 }
ipRouteIfIndex OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The index value which
                uniquely identifies the
                next hop
                to reach the destination
                network."
        ::= { ipRouteEntry 2 }
ipRouteMetric1 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The primary routing
        metric for this route."
        ::= { ipRouteEntry 3 }
ipRouteMetric2 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "An alternate routing
        metric for this route."
        ::= { ipRouteEntry 4 }
ipRouteMetric3 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "An alternate routing
        metric for this route."
        ::= { ipRouteEntry 5 }
ipRouteMetric4 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "An alternate routing
        metric for this route."
        ::= { ipRouteEntry 6 }
ipRouteNextHop OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP address of the
        next hop of this route."
        ::= { ipRouteEntry 7 }
ipRouteType OBJECT-TYPE
        SYNTAX INTEGER {
                other(1), -- none of the
                following
                invalid(2), -- an
                invalidated route
                -- route to directly
                direct(3), -- connected
                (sub-)network
                -- route to a non-local
                indirect(4) --
                host/network/sub-network
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The type of route."
```

```
        ::= { ipRouteEntry 8 }
ipRouteProto OBJECT-TYPE
        SYNTAX INTEGER {
                other(1), -- none of the
                following
                local(2), -- entries
                netmgmt(3), --
                management protocol
                icmp(4), -- e.g.,
                Redirect
                egp(5),
                ggp(6),
                hello(7),
                rip(8),
                is-is(9),
                es-is(10),
                ciscoIgrp(11),
                bbnSpfIgp(12),
                ospf(13),
                bgp(14)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The routing mechanism
                via which this route was
                learned."
        ::= { ipRouteEntry 9 }
ipRouteAge OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of seconds
                since this route was
                last
                updated or otherwise
                determined to be
                correct."
        ::= { ipRouteEntry 10 }
ipRouteMask OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Indicate the mask to be
                logical-ANDed with the
                destination address
                before being compared to
                the
                value in the ipRouteDest
                field."
        ::= { ipRouteEntry 11 }
ipRouteMetric5 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "An alternate routing
        metric for this route."
        ::= { ipRouteEntry 12 }
ipRouteInfo OBJECT-TYPE
        SYNTAX OBJECT IDENTIFIER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A reference to MIB
                definitions specific to
                the
                particular routing
                protocol which is
                responsible
                for this route, as
                determined by the value
                specified in the route's
                ipRouteProto value."
        ::= { ipRouteEntry 13 }
```

Appendix III

ASN.1 BER Encoding

FORMAT for any data of any type (a data element)
T L V (Type octet, Length Octet, value octets)

TYPE OCTET (if only 1 octet)
An ASN.1 definition states a type as [class number] IMPLICIT Type
Only the Type is mandatory.

Type octet = Class + P/C + number

Class is the

00 UNIVERSAL	00 (hex)
01 APPLICATION	40 (hex)
10 Concext Specific (no class stated)	80 (hex)
11 Private	C0 (hex)

P/C is the constructor bit

Primative (a base type)	00 (hex)
Constructor (value contains data elements)	20 (hex)

Number is state in definition but remeber in definition it is decimal.
The following are Numbers for some UNIVERSAL class, primitive types

BOOLEAN	01
INTEGER	02
BIT STRING	03
OCTET STRING	04
NULL	05
OBJECT IDENTIFIER	06

The following are Numbers for some UNIVERSAL class, constructor types

SEQUENCE/SEQUENCE OF	10

LENGTH OCTET is length of data in octets (assumed < 128)

VALUE OCTETS encoded depends on type

BOOLEAN	TRUE is FF (hex), FALSE is 00 (hex)
INTEGER	2's complement in minimum number of octets required
OCTET STRING	Sequence of octets
NULL	Has no contents, length is 0
OBJECT IDENTIFIER	n1.n2.n3. --- .nm
	If all numbers in OID ni < 128, then as follows
	Octet 1 is 40*n1+n2
	Octet 2 is n3
	...
	Octet m-1 is nm

SEQUENCE/SEQUENCE OF
Value is a sequence of data elements each a TLV

ASCII Code Table

ASCII	Hex	Symbol
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
1109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	-
127	7F	