BCS HIGHER EDUCATION QUALIFICATIONS
Level 4 Certificate in IT

March 2012

EXAMINERS' REPORT

Computer & Network Technology

Overall comments

There has been a significant improvement in the answers produced by candidates during this examination session. Overall, candidates were prepared for the various topics examined. It was encouraging to note that centers have taken on board the suggested solutions presented during previous examinations.

Some candidates seem ignore the amount of marks allocated to questions in section A and B. When writing answers, candidates must carefully keep in mind that question A1, A2, A3 and A4 carry 30 marks each. Accordingly, candidates must write sufficiently in-depth answers to attract these marks. Short answers which lack depth did not enable candidates to score well in section A.

A1. a) In the context of computer input/output, what is DMA and what are the advantages and disadvantages of using DMA?

(6 marks)

b) Explain what we mean by the expression *interrupt-driven I/O* and briefly explain what the advantages and disadvantages of interrupt-driven I//O are.

(6 marks)

c) Typical microprocessors like the Intel Pentium family and the 68K family use the system stack to support nested interrupts. What are nested interrupts and how is the stack used to support nested interrupts?

(6 marks)

d) Explain how an interrupt is received from a peripheral and then processed. You are required to use one or more diagrams in your explanation.

(12 marks)

Indicative answer pointers

a)

DMA means direct memory access. It is a form of input/output in which the computer stops processing, gives up the bus (data, address and control bus), and data is transferred directly between the peripheral and memory.

In order to perform DMA, a DMA controller requests the system bus from the CPU. The CPU gives up the system bus and the DMA controller then takes over the system bus and generates the sequential addresses of the source of data (memory to peripheral) or the destination of the data (peripheral to memory). In other words, the DMA controller is a dedicated device that transfers data to and from memory and a peripheral.

Note that some forms of DMA are interleaved; that is they take place when the CPU is not actively using the bus. In this case it is not necessary to halt the CPU during a DMA operation.

The advantage of DMA is its speed. Data can be transferred very rapidly because the DMA controller is hardwired to generate sequential addresses and to perform data transfers using dedicated hardware. However, DMA is more expensive to implement (although DMA control is built into the controllers of modern high-performance buses). One problem that can arise with DMA is the need to inform cache controllers that the memory used by DMA is not available for caching, or to invalidate all cache entries for the memory region using the DMA.

b)

Interrupt-driven I/O is a form of input/output transaction that is triggered by the device taking part in the transaction.

Without interrupt-driven I/O any program that needs data (or wished to output data) would have to "ask" or poll the input or output device to determine whether it was ready or not. This is very inefficient in a computer that can execute instructions at the rate of a billion per second when it is operating in conjunction with an I/O device capable of processing, say, only millions of bytes per second.

Interrupt-driven I/O allows the computer to perform other tasks until a device is ready to take part in I/O. The I/O device interrupts the computer by asserting an interrupt request line. The processor receives the interrupt request and responds to it by executing an interrupt handling program that deals with the source of the interrupt. After dealing with the source of the interrupt, the processor returns to the task it was executing before the interrupt. In other words, interrupt-driven I/O is input or output on demand – it is as if the I/O peripheral jammed a subroutine call into the code when the peripheral was ready.

The advantage of interrupt-driven input or output is that it is efficient. It avoids polling loops and waiting for an I/O transaction to occur. Moreover, it can be turned off (temporarily) to defer to higher interrupt priorities; that is, requests from a keyboard, mouse or printer can be made to defer to interrupt requests from a disk which is a higher-priority device.

Interrupt-driven I/O is slightly more complicated that memory-mapped (i.e., polled) I/O. However, its only real disadvantage is that in a complex system with many I/O devices, it is possible that some low-priority devices may have to wait for a long time before they are serviced.

c)

A nested interrupt is an interrupt that takes place within another interrupt; that is, it occurs when an interrupt is itself interrupted. Consider the following example where an interrupt from device Q is nested within an interrupt from device P:

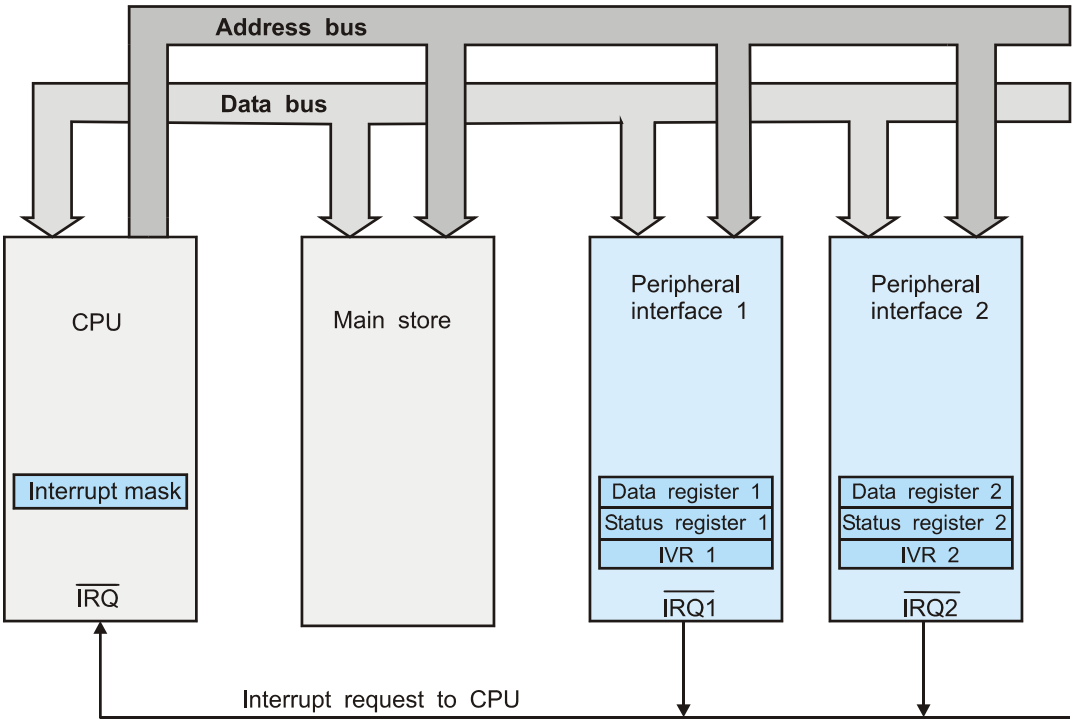
Time t1 Background processing
Time t2 Interrupt from device P is received and interrupt processing for device P begins
Time t3 Interrupt from device Q is received. Device Q has a higher priority than P and P's interrupt handling is suspended in favour of Q. Device Q's interrupt is now processed.
Time t4 Device Q's interrupt processing is finished. Device P's interrupt processing is now resumed.
Time t5 Device P's interrupt processing is finished and the background processing resumed.

The key to nested interrupts (and nested subroutines) is the stack which is a first-in-first-out queue. Interrupt (and subroutine) return addresses are pushed on the stack. Consequently, the return address at the top of the sack represents the most recent return address. Using the above example, the stack is (top of stack on the left).

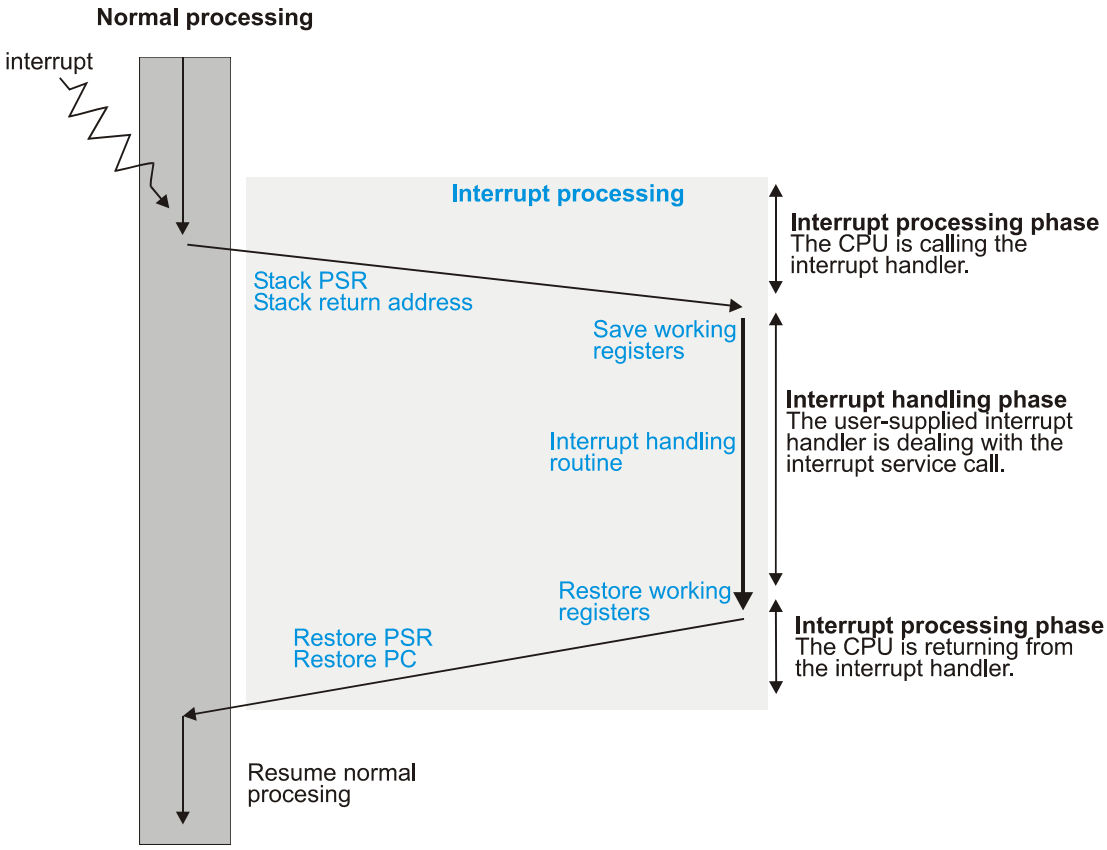
Time t1 empty
Time t2 return to background address
Time t3 return to device P interrupt address, return to background address
Time t4 return to background address
Time t5 empty

d)

The following two figures show (first) the basic interrupt hardware and (second) the interrupt handling sequence. Students are not expected to show diagrams this precise but should give an approximate example of one or both.



The simple hardware (above) uses an interrupt request line from each device capable of requesting an interrupt. When the processor detects that an interrupt has been requested, it compares its level with the value in its interrupt mask. If the interrupt request is at a higher level than the current mask level, the interrupt is processed. The interrupting device is identified by polling each peripheral (i.e., reading its interrupt status register) in turn until the device located. Then the interrupt handling routine is executed.



The diagram above demonstrates interrupt processing. When an interrupt is accepted, the return address and system status s pushed on the stack. The appropriate interrupt handler is called and the interrupt processed. Then the machine status and return address is pulled off the stack. An interrupt is very much the same as a subroutine with two differences. First, the machine status is saved as well as the return address. Second, the target address of the interrupt handler is built into the processor's hardware (or is supplied by the interrupting device) whereas a subroutine call provides the target address as part of the instruction.

A2. a) Low-level instructions can access data in either a memory location or a register. What are the advantages and disadvantages of using:

i) a register to store data and
ii) a memory location to store data?

(2 x 3 marks)

b) Briefly describe the format of an assembly language instruction.

(4 marks)

c) What addressing modes are supported by typical low-level languages?

(6 marks)

d) A computer instruction has a 16-bit format and all instructions are 16 bits long. This machine implements instructions like ADD r1,r2,r3 where r1, r2, and r3 are registers. If the processor implements 16 different instructions, how many registers does the computer have?

(7 marks)

e) Forty years ago, programming computers in assembly language was quite common. Today, it is relatively rare. Why is assembly language an unpopular means of programming computers and why is it less popular today than 40 years ago?

(7 marks)

Indicative answer pointers

a)

(i) A register and a memory location perform exactly the same function – they both save a word of data. The difference is that the register is on-chip, has an access time comparable with the processor's clock cycle time, and does not require a long address (if there are 32 on-chip registers, a register requires a 5-bit address to access it). Consequently, registers are the preferred form of data storage. Unfortunately, registers are expensive (in terms of the hardware cost) and only a few can be located on the CPU chip (partially because of the address restriction – number of address bits available to specify a register) and partially because of the cost of fabricating on-chip registers.

(ii) Memory is very cheap; for example, a typical PC might have 4GB (2^{32}) bytes of main store. Unfortunately, memory is very slow compared to registers (today, an order of magnitude or slower). Moreover, because memory locations typically require a 32-bit address, it is necessary to use a pointer to access memory – this further slows access.

b)

There are many possible formats of an assembly language instruction and any sensible answer that discusses the fundamental issues was acceptable.

Assembly language instructions are generally the same width as the computer's basic word. Because computers falling into the CISC category (like the Intel Pentium family and iCore family) were derived from 16-bit processors, the basic instruction word is 16 bits or multiples of 16 bits. RISC processors (ARM SPARC, MIS) generally have fixed-length 32-bit instructions.

CISC data processing instructions are frequently of the form ***operation, source address, destination address***. Because only two addresses are specified, and many operations are dyadic (requiring two operands), one address provides a source operand and the other address provides a second source address and a destination operand; for example ADD A,B means A = A + B. Generally, CISC processors use a 1 ½ address format in which one operand is an on-chip register and the other address is a memory location.

CISC instructions use register-to-register operations of the form ADD r1,r2,r3 where r1 = r2 + r3. These computers have a simple load and a simple store instruction – both of which are pointer based; for example LDR r2,[r1] means load register r2 with the data pointed at by r1.

As well as data processing and register load/store operations, other types of instruction are branch and subroutine call (e.g., BEQ target or BSR function), and test (e.g., CPM D3,#4 that compares the contents of a register with a literal)

c)

There are three fundamental addressing modes:

i) Literal – an operand is an actual value that is part of the instruction. These are used where the operand is known at compile time (i.e., it's a constant). Typical examples are in loop counters or comparisons. For example, MOV r1,#4 (move 4 into r1) or CMP r2,#3 (compare r2 with 3) or ADD r1,r1,#1 (add 1 to r1).

ii) A memory address where the location of an operand is specified; for example, ADD D1,1234 means add the contents of D1 and memory location 1234 together and put the result in location 1234.This addressing mode is used by CISC processors to deal with variables (i.e., values in memory). This addressing mode is not implemented by RISC processors

iii) Pointer-based or register indirect addressing where the location to be accessed is in a pointer or index register. This is the only memory-based addressing mode available to RISC processors. A typical operation is LDR r1,[r3] which means load register r1 with the contents of memory pointed at by r3. Since r3 contains a variable that can be accessed by other instructions, this addressing mode can generate variable addresses at run time and be used to access data structures such as arrays, tables and lists.

d)

The number of bits in an instruction is made up of instruction op-code field plus addressing fields

If there are 16 different instructions, the number of bits used to select an instruction is 4 since $2^4 = 16$

The number of bits devoted to register selection is instruction length – opcode size = 16 – 4 = 12

Three registers have to be specified (in this case r1,r2,r3)

The number of bits used to select each register is 12/3 = 4

The maximum number of registers is $2^4 = 16$.

e)

Four decades ago, there were fewer software development tools available to the programmer. Sometimes, there was little alternative to the use of assembly language (particularly, for those developing applications using 8-bit microprocessors). Today, there is a far wider range of development tools and there are more high-level languages. In particular, there are more high-level languages that provide high levels of programmer productivity via either features of the language or library features that are accessibly by the programmer.

Assembly language is not easy to use to write large programs – it is possible to write relatively small fragments of assembly language code, but human programmers find it harder to deal with large assembly language programs. Consequently, the productivity of an assembly language programmer can be lower than a high-level language programmer.

Assembly language programs are very hard to debug and maintain (especially by those who did not write the original code).

Assembly language programs sometimes rely on “tricks” or unusual coding practice to best exploit the code – this is difficult to do and difficult (for another programmer) to follow.

Some modern processors have rather complex instruction sets that make it very difficult to write in assembly language – for example the Intel Itanium processor.

A3 a) Computers are constructed from two types of logic elements, gates and sequential elements (e.g., flip-flops). What are the characteristics of these two classes of element and why are two different types of element needed to construct a computer? Your answer should include a description of the basic gates and a description of the type of sequential circuits used in computers.

(12 marks)

b) A circuit has four natural binary encoded inputs D, C, B, A where D is the most-significant bit. These values represent 0 to 15 in decimal. It has a single output F.

F is 1 if the input on D, C, B, A is in the range 3 to 7 (inclusive) or 12 to 15 (inclusive).

Construct a truth table for this system.

(6 marks)

c) Obtain a simplified Boolean expression for F using any suitable technique.

(6 marks)

d) Construct a logic circuit for the above system using AND, OR, and NOT gates that will provide a 1 output if the input conditions are met.

(6 marks)

Indicative answer pointers

(a) Combinational elements or gates are used to construct circuits whose output is a Boolean function of the current inputs. If the inputs are a,b,c,d,… the output is given by f(a,b,c,d,…). These are called memory-less circuits because the output does not depend on any previous state or past history of the circuit.

There are three fundamental gates, AND, OR and NOT. The AND gate can have n inputs and its output is true if and only if all n inputs are true. The OR gate can have n inputs and its output if true if one or more inputs are true. The NOT gate has one input and its output is the inverse/complement of the input and is false if the input is true, and true if the input is false.

All digital circuits can be constructed from these three gates alone.

However, if an AND gate is followed by a NOT gate in series, the new gate is called a NAND gate. It is possible to synthesis a NOT gate, and AND gate and an OR gate using NAND gates only. Therefore all digital circuits can be constructed from NAND gates. The same is true of the NOR gate (an OR gate followed by a NOT gate).

Boolean logic circuits are used to synthesize all the logic in a computer (arithmetic logic, adders, multipliers, multiplexers, demultiplexers, coders and decoders, majority logic circuits, priority circuits, voting circuits, and so on).

The computer executes instructions one by one sequentially. This involves the element of time. The other class of circuit is the sequential circuit that is constructed from flip-flops or bistables. These are logic elements whose output is a function of the current input and the past history of the circuit. That is, if the inputs are a,b,c,d,… and the previous output is Q_{old}, then the new output is $Q_{new} = f(a,b,c,d,..,Q_{old})$. Typical fundamental sequential circuits are D, RS, T, and JK flip-flops. As in the case of gates, any flip-flop can be used (with additional gates) to synthesize any other flip-flop. Moreover, all flip-flops can be constructed from basic gates alone.

A D flip-flop is a circuit with a D (data) input and a clock, C. When the D flip-flop is clocked, the data at the D input is transferred to the Q output and held until the next time it is clocked. That is, the D flip-flop is a 1-bit memory. The RS flip flop has two inputs R (reset) and S (set). When S is asserted, the Q output goes to 1 and remains at 1 until the R input is asserted. Like the D flip-flop, this is a 1-bit memory. The JK flip-flop has properties of the RS flip-flop plus the additional property that if inputs J=K=1, the output Q toggles or changes state each time it is clocked.

Flip-flops are used to create memory elements (registers), shift registers that move strings of bits left or right (to perform binary multiplication by 2 or division by 2), and counter that count through a sequence of states (e.g., a binary up counter that counts 0,1,2,3,4… or a sequence generator that can step through any arbitrary sequence such as 4,7,1,3,9,12,2).

Sequential plus Boolean elements make is possible to design any digital machine.

A computer can be considered to consist of a counter (program counter that steps through the instructions), a memory (an array of flip-flops), registers (flip-flops), and an ALU (Boolean combinational logic).

b)

Truth Table

D	C	B	A	Number	3 to 7	12 to 15	F
0	0	0	0	0			0
0	0	0	1	1			0
0	0	1	0	2			0
0	0	1	1	3	Yes		1
0	1	0	0	4	Yes		1
0	1	0	1	5	Yes		1
0	1	1	0	6	Yes		1
0	1	1	1	7	Yes		1
1	0	0	0	8			0
1	0	0	1	9			0
1	0	1	0	10			0
1	0	1	1	11			0
1	1	0	0	12		Yes	1
1	1	0	1	13		Yes	1
1	1	1	0	14		Yes	1
1	1	1	1	15		Yes	1

c)

From the table

$F = \overline{D}.\overline{C}.B.A + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + \overline{D}.C.B.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.A + D.C.B.\overline{A} + D.C.B.A$

Using Boolean algebra

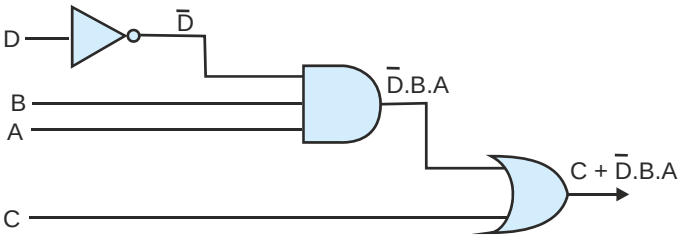
$F = \overline{D}.\overline{C}.B.A + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + \overline{D}.C.B.\overline{A} + D.C$

$F = \overline{D}.\overline{C}.B.A + \overline{D}.C + D.C$

$F = \overline{D}.\overline{C}.B.A + C$

$F = \overline{D}.B.A + C$

d)



A4. The greatest problem facing both domestic and commercial computers is that of malware. Malware is a term that describes all programs that are intentionally designed to harm the computer user either directly or indirectly. SPAM is included in this category, although it is it is not a program.

Write an essay on the range of malware available and the type of problems created by the various types of malware. Your answer should include the dangers and effects of malware, the reason computers are vulnerable to malware, and a discussion of the things that the computer user can do to minimize both the dangers of malware and the consequences of malware.

(30 marks)

Indicative answer pointers

This is an open-ended question. In order to receive a pass mark, students were expected to provide a basic but coherent description of at least three types of malware, how they are used, and what threats they present. Students were also expected to indicate some of the steps that can be taken to protect against malware. In order to get a higher mark, students needed to list a wider range of malware and provide a more in-depth answer with greater detail.

Examples of malware

SPAM or junk mail. SPAM is unsolicited e-mail that is usually sent in bulk (i.e., the sender transmits many – often millions – of messages to recipients). Typically, SPAM provides unsolicited advertising. It does not present a danger as such to the computer, but can lead to dangers (inviting the user to buy illegal content, illegal drugs, or illegal services). Moreover, it can lead to traps (fraud that attempts to get your bank details). One of the main dangers of SPAM is that it fills your mailbox, takes up valuable bandwidth, and can lead to genuine email being lost. SPAM can be avoided (to some extent) by keeping your email address secret (i.e., not publicizing it) and by using filters.

SPAM can partially be prevented by the use of email filters and firewalls. However, the tighter the filter (i.e., the SPAM detection threshold) the greater the danger that legitimate messages will be wrongly categorized as SPAM.

ADWARE

Adware is related to SPAM and is, normally, a non-harmful category of malware in the sense that its purpose is not to harm the computer but to sell a product. Adware can be incorporated in

software that is installed on a computer and its function is to sell products – often by means of pop-ups that appear on the screen. Adware is strongly associated with freeware because it provides a means of generating income to those who distribute software at no cost. It can be avoided by carefully controlling the software installed on a computer. Commercial programs can be bought to search a computer of instances of adware and to delete them.

SPYWARE

Spyware is closely related to adware but is potentially far more dangerous. Spyware is installed in the same way as adware and, unlike a virus, is not designed to propagate itself or to harm a computer. Spyware is intended to report back to the originator. Such reports can be anything from the user's internet browsing activities (for use by targeted advertising) to details of their bank account (for use by those who wish to perform financial theft or even identity theft). Spyware can also consume CPU and disk resources by searching files and sending data to the spyware originator.

Spyware can be prevented in the same way as adware – by means of controlling you computer use (the installation of software and careful use of the internet). Commercial packages can be installed to detect spyware.

VIRUSES and ROOTKITS

The computer virus is one of the most potentially dangerous examples of malware. A virus is so-called because it mimics the biological virus by replicating itself. A virus may or may not carry a payload (there is a distinction between the propagation mechanism of a virus and its intention or payload). The payload of a virus may be a simple message. Indeed, it is possible to have a good virus that spreads updates and corrections to software. However, viruses can have a payload that caused harm by erasing or corrupting data. A virus could selectively corrupt many files over a long period because the virus was not detected.

Viruses initially spread via programs copied from floppy disks. Today, they are invariably web/Internet based.

Viruses must be able to take control of a computer which means that they must be able to execute code. Some viruses are called macro viruses because they exploit macros used by software such as word processors and spreadsheets. Some viruses exploit features of computer languages (for example, the lack of bounds testing on some languages that allow you to inject code into a data area).

Viruses can be avoided by careful security (i.e., loading only programs known to be safe). Today, there are many commercial antivirus packages that monitor computer activity for evidence of viruses. However, none of these packages are perfect and there is a continual war between virus writers and anti-virus software producers. Viruses are often designed to mutate and to change their signature (the sequence of code bytes used by a virus detector to locate a given virus). It is suggested that non-changeable media such as optical storage be used to backup data because the information stored cannot be modified by a virus. Care should be taken with transportable storage such as flash drives because viruses have been designed to exploit them (e.g., a person at a conference may use an infected flush drive on the conference computer and infect all the other delegates using that computer to give their presentation).

A rootkit (or root kit) is a relatively modern form of virus that operates at the root of a directory and is a particularly difficult form of virus to detect and remove.

WORM

A worm is a form of SPAM and is not, generally, dangerous (harmful to individual computers). Like a virus, a worm spreads from computer to computer. Unlike a virus it is not designed to attach itself to software and cause harm. The worm is aimed more at the network by which it spreads than the target computer. A worm consumes network bandwidth as it spreads through the network. Worms can be considered as a mechanism that creates a *denial of service*; that is, they can be considered as an attack on a network. Worms often propagate by exploiting weaknesses on computer operating systems. Therefore, to some extent, protection against worms lies with the operating system writers.

Worms can also be used to propagate SPAM. A worm affects a computer and then uses the infected computer to send out SPAM (making it difficult to trace the originator of the SPAM). As in the case of most other malware, commercial programs exist that detect worms and remove them infected systems.

TROJAN

The Trojan is a reference to the Trojan Horse in which soldiers were smuggled into Troy in a giant wooden horse disguised as a gift. In the computer world, a Trojan is a piece of software that contains malware. Trojans are associated with freeware and other widely available downloadable software. Once loaded, the software can perform the function dictated by the malware writer. Typically, Trojans are used to monitor a system (spyware) or to propagate advertising (adware). As in the case of most malware, protection lies in vigilance (not loading suspect software – checking software for malware by looking it up on the Internet – using commercial programs to check for it).

Section B

B5. The following programs are used to detect networking problems. Briefly describe how each program works.

a) Ping **(4 marks)**

b) Traceroute **(4 marks)**

c) Ipconfig **(4 marks)**

Indicative answer pointers

a) A utility to determine whether a specific IP address is accessible. It works by sending a packet to the specified address and waiting for a reply. PING is used primarily to troubleshoot Internet connections. There are many freeware and shareware Ping utilities available for personal computers.

b) A utility that traces a packet from your computer to an Internet host, showing how many hops the packet requires to reach the host and how long each hop takes. If you're visiting a Web site and pages are appearing slowly, you can use traceroute to figure out where the longest delays are occurring.

c) A utility which provides IP address of devices on the network. Various pieces of information can be obtained. It also allows various commands to be issued to devices and adapters on the network.

B6. Memory management is vital in computer systems. Describe each of the following types of memory and explain how it is used.

a) Cache memory

(6 marks)

b) Virtual memory

(6 marks)

Indicative answer pointers

a) A special high-speed storage mechanism. It can be either a reserved section of main memory or an independent high-speed storage device. Two types of caching are commonly used in personal computers: *memory caching* and *disk caching.*

b) The purpose of virtual memory is to enlarge the *address space*, the set of addresses a program can utilize. For example, virtual memory might contain twice as many addresses as main memory. A program using all of virtual memory, therefore, would not be able to fit in main memory all at once. Nevertheless, the computer could execute such a program by copying into main memory those portions of the program needed at any given point during execution.

B7. Mobile networking has become very popular among computer users.

a) Explain what a WLAN is.

(4 marks)

b) Describe the wireless networking standards IEEE 802.11x

(8 marks)

Indicative answer pointers

a) Wireless Local Area Network - A type of local-area network that uses high-frequency radio waves rather than wires to communicate between nodes.

b) 802.11 and *802.11x* refers to a family of specifications developed by the IEEE for w*ireless* LAN (WLAN) technology. 802.11 specifies an over-the-air interface between a wireless client and a base station or between two wireless clients. The IEEE accepted the specification in 1997

B8. An operating system provides key functions to a computer system. Using suitable examples, explain the following operating systems related terms:

a) Graphical User Interface

(4 marks)

b) Mobile operating systems

(4 marks)

c) System Boot Process

(4 marks)

Indicative answer pointers

a) Graphical User Interface enables users to issue commands to computers and related devices without the need to types these. Windows, Icons through the use of a Mouse enable

instructions to be issued. It's made the use and manipulation of data and instructions simpler.

b) Mobile Operating System refers to an operating system designed for mobile devices such as mobile phones, tablets and PDAs. This type of OS enables devices to carry various functions such as accessing the internet and using applications.

c) Boot process enables a computer and other related devices including mobile phones and PDAs to start operation. The process enables the loading of relevant programs in the main memory so that various tasks can be carried out.

B9. a) Distinguish between the operation of a laser and an ink jet printer.

(8 marks)

b) Explain the factors which influence the choice between the above printers.

(4 marks)

Indicative answer pointers

a) An ink jet printer produces output using an ink cartridge through a nozzle type device. A laser printer produces output by the direct scanning of a laser beam across the printer's photoreceptor.

b) Issues such as speed, volume of output, running cost, etc.

B10. Differentiate between:

a) Hub

(4 marks)

b) Switch

(4 marks)

c) Router

(4 marks)

Indicative answer pointers

a) A common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets.

b) A device that filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model and therefore support any packet protocol.

c) A device that forwards data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network. Routers are located at gateways, the places where two or more networks connect. Routers use headers and forwarding tables to determine the best path for forwarding the packets.

B11. Data communications can be performed using different technologies. Describe the differences between PSTN and VoIP and explain why each of the technologies is used.

(12 marks)

Indicative answer pointers

To understand how VoIP, short for Voice over Internet Protocol, works, it is helpful to compare it to how conventional phone calls operate. When you place a "regular" phone call using the Public Switched Telephone Network (PSTN), also known as Plain Old Telephone Service (POTS) you use what's called circuit-switched telephony. This system works by setting up a dedicated channel (or circuit) between two points for the duration of the call. These telephony systems are based on copper wires carrying analog voice data over the dedicated circuits.

This is in contrast to newer Internet telephony networks based on digital technologies. VoIP, in contrast to PSTN, uses what is called packet-switched telephony. Using this system, the voice information travels to its destination in countless individual network packets across the Internet.

B12. Compare and contrast each of the following data storage devices. Explain how each is used.

a) RAM **(3 marks)**

b) USB drive **(3 marks)**

c) Magnetic Disk **(3 marks)**

d) Hard disk **(3 marks)**

Indicative answer pointers

a) RAM – volatile memory which enables data to be temporarily stored on the computer. This type of memory enables the immediate tasks of the computer to be performed.

b) USB drive- this provides an external mean of storing data. The USB disk can be used to large volume of data and carry this between devices.

c) Magnetic Disk – most common way of storing data using magnetic medium. The disk store data in tracks and sectors. A magnetic disk drive is used to write data to and read data from the surface of the disk.

d) Hard disk – this disk provides permanent storage of large volume of data on a computer system. The hard disk holds items of software which are needed to run various applications.