# 2012 Computing

# Advanced Higher

# Finalised Marking Instructions

**SECTION I – Software Development**

*Marks*

**1.** **(a)** **(i)** Technical feasibility **1 KU**

**(ii)**
- Processor power – to process game instructions quickly enough for game play.
- Amount of RAM – to hold the game and the game state including other users' data.
- Graphics card/capability – to render graphics in reasonable time.
- Range/bandwith of wireless – high enough so that there is no lag in the game.
- Any other reasonable.

**2 PS**

**Justification must include context.**

**(b)** **(i)** Clarifies detailed and exact descriptions of the features to be present in the proposed software. **1 KU**

**(ii)**
- Analysed at the beginning of the design stage to identify the functions to be planned.
- Used to validate other stages or analysed at the end of design/implementation/etc to ensure functional requirements are being met.
- Dispute resolution between client and developer.

**1 KU**

**(iii)**
- Includes scope boundaries which detail the limits that the program will handle/clarify what the proposed software will not do.
- Inputs/outputs are specified.

**1 KU**

**1 mark any valid**

**(c)**
- Setting of timescales/deadlines.
- Allow identification of concurrent tasks.
- Allow identification of the dependency of tasks on the completion of others.
- Allow you to view percentage complete.
- Use Gantt/Pert chart.
- Any other reasonable.

**2 KU**

**(d)** **(i)**
- step through the code **(1)**
- recording changes to variables **(1)**.

**2 KU**

**(ii)**
- Watch a variable **(1)** allowing you to inspect its contents during execution. **(1)**
- Set breakpoints **(1)** to stop execution of the code at specified lines of code. **(1)**
- Dry run **(1)** – manually stepping through the lines of code. **(1)**

**2 KU**

***Marks***

**2.** **(a)** **(i)**

- Table/grid of values
- with row and column indexes
- all of the same type.
- An array of 1d arrays.

**2 KU**

**Max 2 marks**

**(ii)** Puzzle_state(6,6) as integer
Puzzle_state(5,5) as integer

**2 PS**

Syntax may vary.
**1 mark for two matching indices, 1 mark for data type**

**(b)**

```
Set errors to 0
Loop for rows (1 with end)
        Loop for columns (1 with end)
                If Puzzle_state (rows,columns) does not equal
                solution (rows, columns) then
                        Errors=errors +1 (1 for increment and
                        initialisation)
                End if
        Next columns
Next rows
```

**4 PS**

**1 for indication of 2d array**

**(c)** Testing the integration of subroutines/units tested as a group **(1)** using

- black-box testing **OR**
- test cases to check units interact together correctly/parameter passing is correct **OR**
- top-down or bottom-up approach.

**2 KU**

Beta testing **(1)** with

- end users providing feedback **OR**
- end user systems providing error reports.

**Max 2 marks. Allow two descriptions from the same type of testing.**

| | | | ***Marks*** |
|---|---|---|---|
| **3.** | **(a)** | People(5 000 000) **(1)** as memberDetails **(1) OR**<br>An array **(1)** of memberDetails **(1)**<br>People:array[1..5000000] **(1)** of memberDetails **(1)**. | **2 PS** |
| | **(b)** | Linear search compares target name <u>in sequence</u> **(1)** starting from lowest index and so names at the start of the array will be found much quicker than those at the end **(1)**. | **2 PS** |
| | **(c)** | Password is not sorted/Username is sorted **(1)**.<br>Password has multiple entries that are the same/username are unique **(1)**. | **2 PS** |
| | **(d)** | Ask for search_item<br>Set low to 1 (lowest index)<br>Set high to 5000000 (highest index)<br>Start Loop<br>Set mid to (low+high) div 2<br>If search_item<people(mid).username then<br>High = mid-1<br>Else<br>Low = mid+1<br>End if<br>Loop until search_item=username(mid) or low>high<br><br>**1 mark for both initialisations**<br>**1 mark with loop and matching termination**<br>**1 mark for setting middle**<br>**1 mark for IF with condition and END IF**<br>**1 mark for high/low being set matching the IF condition**<br>**1 mark for the mid-1/mid+1 to match High/Low** | **6 PS** |

**4.** **(a)** Data fields/attributes/properties (state) variables **(1)** used to store values that give the state of the object. **(1) OR** **2 KU**

Class hierarchy or superclass **(1)** which defines the inheritance. **(1)**

**(b)** **(i)**
- Front =front+1
- Rear does not change

**2 PS**

**(ii)**
- 1 mark for 15 and 29 in array but not in queue
- 1 mark for 8,11 and 9 in queue in correct order
- 1 mark for front=2 and rear=4

**3 PS**

**If candidate has removed 15/29 then award the mark for matching front and rear values.**

**(c)** **(i)** End of array will be reached **(1)** but there is room in the array for items to be placed in the queue **(1)**. **2 PS**

**(ii)** Use wraparound/circular queue where items joining the queue will be added to the start of the array and the pointer rear will become 1 **OR**
Reset the array by shuffling items to the front of the array. **1 PS**

**(d)** **(i)**
- New queue would be a subclass of Queue. **(1)**
- Only additional data fields/variables and methods would need to be defined in the new subclass **(1)**.

**2 KU**

**(ii)**
- As New queue will <u>inherit</u> data and methods from the superclass **(1)** fewer new lines of code need to be written **(1)**
- Reducing the additional lines of code that need to be implemented **(1)** resulting in faster development time **(1)**.
- Reducing the need for testing since only the additional methods need testing **(1)** resulting in faster development time **(1)**.
- Any other reasonable

**2 PS**

***Marks***

**5** **(a)** **(i)**

- Compare 1 and 2, 1 and 3 etc
- Swaps if required
- Repeat from 2 and 3, 2 and 4 etc

**2 KU**

**(ii)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 19 | 18 | 12 | 15 | 5 | 16 | 15 | 14 |
| 19 | 18 | 16 | 12 | 5 | 15 | 15 | 14 |

**2 PS**

**1 mark each row**

**(iii)** 7+6+5+4+3=25 **or** 8*7/2-(2+1) **2 PS**

**1 mark for 25 and 1 mark for working/explanation.**
**Allow 1 mark for 8*7/2 or 7+6+5+4+3+2+1.**

**(b)** **(i)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | 16 | 15 | 19 | 15 | 14 | 12 | 5 |

**1 PS**

**(ii)** Three **(1)** because the worst three marks will be placed at the end in the sorted part of the list **(1)**. **2 PS**

**(c)** **(i)** One. **1 PS**

**(ii)**

- determine when no swaps occur
- allowing the algorithm to terminate
- by using a (BOOLEAN) flag

**2 PS**

**Maximum of 2 marks.**

[END OF SECTION I]

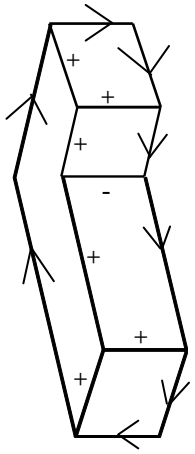
**SECTION II – Part A – Artificial Intelligence**

***Marks***

**6.** **(a)** Convex edge **(1)**. **2 KU**
Concave edge **(1)**.

**(b)** (iii). **1 KU**

**(c)** **3 PS**




**1 mark for 7/8 ⋀ correct**
**1 mark for 5/6 + correct**
**1 mark for – correct**

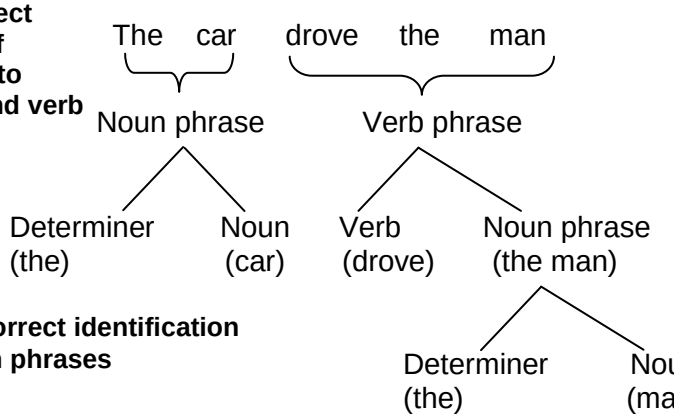
**7.** **(a)** Pragmatic analysis. **1 KU**

**(b)** Proper noun. **1 KU**
Determiner with an adjective and a noun.
Adjective and noun.

**1 mark for one valid**

**(c)** **(i)** **3 PS**

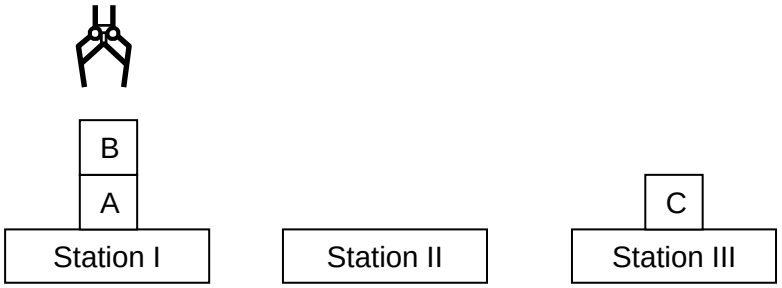
**1 mark for correct identification of primary split into noun phrase and verb phrase**




**1 mark for correct identification of <u>both</u> noun phrases**

**1 mark for correct primary decomposition of verb phrase**

There are other possible parse trees at primary split eg noun phrase/verb phrase (drove)/noun phrase.

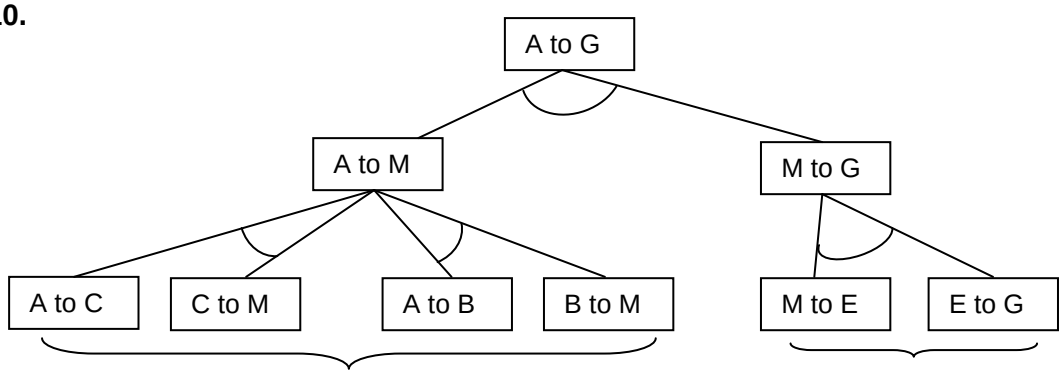
| | | | Marks |
|---|---|---|---|
| | **(ii)** | Concept of **matching words** to lists of nouns/proper nouns/determiners etc for match **(1)**.<br>**OR**<br>**Matching a list of sentence structures**, eg noun phrase + verb phrase/pronoun + verb etc **(1)**.<br>**OR**<br>If no match, backtrack to next rule **(1)**.<br>Other answers are possible. | **2 KU** |
| | **(iii)** | A sentence can pass the syntactical analysis but is nonsense **(1)** and this is picked out by the semantic analysis **(1).** | **2 KU** |
| | **(d)** | Word which can be used as a noun or a verb, eg duck (He saw her duck).<br>**"a word that has 2 meanings" is not enough for the mark.** | **1 PS** |
| **8** | **(a)** |  <br>**1 mark for all items in correct positions** | **1 PS** |
| | **(b)** | ontable(C,II), on(B,C), on(A,B) — **1 mark**<br>clear(A), armempty(lll) — **1 mark**<br>**Order of elements does not matter** | **2 PS** |
| | **(c)** | C is on top of B so B cannot be clear. | **1 PS** |

**9.** **(a)** The system is provided with examples of full, not full and overflowing, and it uses these to make generalisations and form concepts, so that it can then apply these each time a ladle has to be filled. **2 PS**

**1 mark for general explanation of learn by example + 1 mark for relating to scenario**

**(b)** The situation changes each time the ladle is used **(1)**, rote learning needs repeated situations **(1)**. **2 PS**

**10.** **4 PS**




**1 mark for two alternative routes correctly identified**
**1 mark for 'ANDs' identified and annotated correctly**
**1 mark for OR identified and annotated correctly**

**1 mark for correct identification and annotation**

| | | | | ***Marks*** |
|---|---|---|---|---|
| **11.** | **(a)** | | Start state: (0,4,2) **(1)**<br>Goal state: (3,3,0) **(1)** | **2 PS** |
| | **(b)** | **(i)** | Fill 2 litre jug from 5 litre jug<br>Fill 2 litre jug from 4 litre jug } **1 mark**<br>Empty contents of 4 litre jug into 5 litre jug<br>Empty contents of 2 litre jug into 5 litre jug } **1 mark** | **2 PS** |
| | | **(ii)** | l>=4-m **OR** there must be enough in the 5 litre jug to fill the 4 litre jug **(1)**.<br>m<4 **OR** the 4 litre jug cannot be already full **(1)**. | **2 PS** |
| **11** | **(c)** | **(i)** |  | **1 PS** |
| | | **(ii)** | Empty contents of 2 litre jug into 4 litre jug. | **1 PS** |
| | **(d)** | **(i)** | 3 | **1 PS** |
| | | **(ii)** | 0 | **1 PS** |
| | **(e)** | **(i)** | At level 1 select node (2,4,0) then select (5,1,0) but then stops **(1)** since there is no value of this branch on level 3 with a value higher than 3 **(1)** | **2 PS** |
| | | **(ii)** | Best-first. | **1 PS** |
| | | | **Note: A* is not suitable in this example.** | |

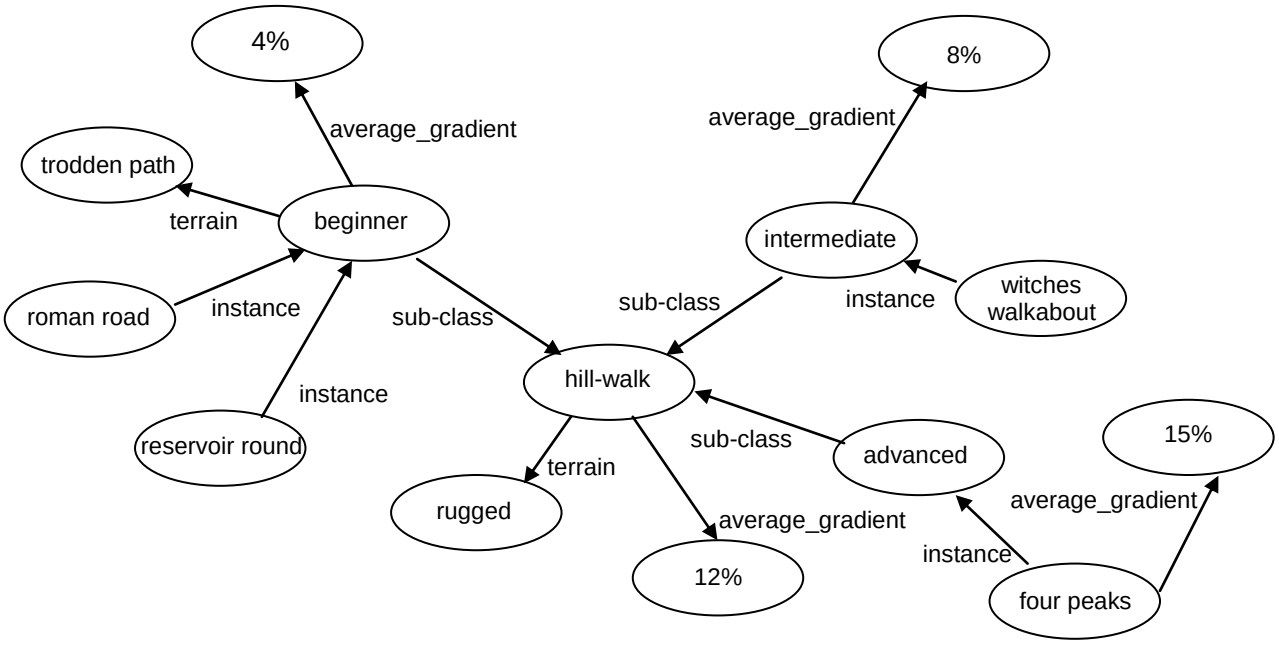
**12.** **(a)** To inform the needs of the system being developed, eg backing storage/RAM requirements… **1 KU**

**(b)** **(i)** A **single/individual** example of a sub-class. **1 KU**

**(ii)** A group of objects which: **1 KU**
share characteristics unique to the sub-class **(1)**
**OR**
inherit characteristics from its superclass. **(1)**

**(iii)** The class has this value, which is inherited by sub-classes and instances **(1)**, unless over-written by an actual value **(1)**. **2 KU**

**(c)** **4 PS**




**2 marks for all nodes – (allow 1 to be missing)**
**1 mark if no more than 3 nodes missing**

**1 mark for all 4 predicate types used correctly at least once**
**1 mark for arrows shown correctly on each of the four predicates**

**(d)** **(i)** terrain(X,Y) if instance(X,Z), terrain(Z,Y) **(1)** **2 PS**
average_gradient (X,Y) if instance(X,Z), average_gradient (Z,Y) **(1)**

**(ii)** One rule needs to be created to allow terrain to be inherited **(1)**. **2 PS**
A second rule needs to be created to allow average_gradient to be inherited **(1)**.
**OR**
There are two properties to inherit **(1)** so one rule needs to be written for each property **(1)**.

**(e)** The search compares the goal value with the head of the list. **(1)** **3 KU**

If it doesn't match, it is then compared with the tail of the list **(1)**. Which is done by repeatedly extracting the head from the tail or splitting the tail. **(1)**

This is repeated until the tail of the list is empty or they find it in the list. **(1)**

**Any 3 from 4**

**(f)** **(i)** The list of the rules which could be triggered (as their conditions so far are all true). **1 KU**

**(ii)** Two from the following: **2 KU**

- **first come first served** fires first rule it comes to whose conditions are all satisfied.
- **recency** fires the rule that uses data that was added most recently.
- size ordering fires the rule with the largest number of the specific conditions.
- data ordering gives **priority** based on information provided by the knowledge engineer.
- context limiting divides the knowledge base into groups of rules which are related to each other.
- explanation of any other valid strategy.

[END OF SECTION II – Part A]

**SECTION II – Part B – Computer Architecture**

*Marks*

**13. (a)**

PC = > MAR 2510 } **1 mark**
Read line pulsed }
Memory = > MDR 57 **1 mark**
PC incremented by 1 2511 **1 mark**
MDR = > IR 57 **1 mark**
Instruction decoded/executed } **1 mark**
A = > 2 }

**+ 1 mark for values in MAR/PC/IR correct**

**6 PS**

**(b) (i)** eg JMP 1920 — Jump to location 1920. **(1 KU, 1PS)**

**(ii)** eg TAX — Transfer Accumulator value into X register. **(1KU, 1 PS)**

**(c) (i)** The code will add up the numbers from 1 to 4 **(1)** and store the total in location 'result' **(1) OR** stores 10 in 'result' **(2)**. **2 PS**

**(ii)** The branch back to 'continue' would not happen **(1)** so 1 would be stored in 'result' **(1)**. **2 PS**

**14 (a) (i)** Level 1 cache is checked to see if it is storing the memory location to be fetched. If it is, a cache hit occurs and the contents are retrieved **(1)**.
If not, a cache miss occurs and level 2 cache is checked to see if it is storing the location. If it is, a cache hit occurs and the page of cache memory containing the value is transferred to a block in level 1 cache from where it is read **(1)**.
If not, a cache miss occurs. A page of main memory containing the location to be fetched is transferred into a page of level 2 cache **(1)** and then a smaller block is copied into level 1 cache from where it is read **(1)**. **4 PS**

**(ii)** Because of the loops, the CPU will spend much of its time executing small areas of code so once this code is stored in cache **(1)** the code will be executed considerably faster since cache is much faster to retrieve code from than main memory **(1)**. **2 KU**

**(b) (i)** When the instruction is executed, the operation is carried out on all of the data items <u>simultaneously</u>. **1 KU**

**(ii)** The operation could be to brighten all pixels on the display. **1 PS**

**1 mark for any valid answer**

**(iii)** The SIMD instruction would be a shift left operation on the 8 registers simultaneously. Each 128 bit register can hold data from four 32 bit pixels **(1)**. There will be a theoretical 4 x 8 = 32 increase in the processing speed for this operation **(1)**. **2 PS**

| | | | | Marks |
|---|---|---|---|---|
| **15.** | **(a)** | | Hard drive issues a transfer request to the DMAC. When CPU does not need to use the buses, it sends an Acknowledge to the DMAC and 'disconnects' from the buses. **(1)** The DMAC now uses the buses to transfer data directly from the hard drive to memory. When transfer is complete or the CPU needs to use the buses again, the DMAC stops transferring data, relinquishes use of the buses and awaits another DMA request. **(1)**<br><br>**(+ 1 mark for diagram)** | **3 KU** |
| | **(b)** | | Without DMA, the data is first moved from the hard drive to a register in the processor before being transferred to the memory. The CPU therefore has to spend its time doing the data transfer. **(1)** With DMA, the data is transferred directly from hard drive to memory, usually whilst the CPU is doing processing tasks that do not need the use of the buses. **(1)** The efficiency of the CPU is therefore considerably improved. **(1)**<br><br>**Any 2** | **2 KU** |
| **16** | **(a)** | **(i)** | RISC: eg a large number of general purpose registers **(1)** to allow more processing within CPU using register oriented instructions improving processing speed because registers are the fastest form of memory **(1)**. | **2 KU** |
| | | **(ii)** | CISC: eg a larger instruction set **(1)** that can reduce programming effort for the programmer **(1)**.<br><br>**Any acceptable answers allowed:**<br>**1 mark feature, 1 mark benefit.** | **2 KU** |
| | **(b)** | **(i)** | When a branch instruction is detected entering a pipeline **(1)**, a second pipeline is used to load code starting at the 'jump' location of the branch while the first pipeline continues to load code from the current PC (non-jump) location **(1)**. When the branch is actually executed in the first pipeline, the CPU will then use the pipeline that contains the correct code whilst the other pipeline is cleared and made available for other use **(1)**. | **3 PS** |
| | | **(ii)** | The CPU can continue processing immediately whichever way the branch goes/there is no pipeline stall. | **1 PS** |
| | **(c)** | | Data flow analysis **(1)**. The assembler rearranges the code to try to avoid situations where parallel processing has to pause because an instruction is dependent on the results of another **(1)**. | **2 PS** |

| | | | | *Marks* |
|---|---|---|---|---|
| **17** | **(a)** | **(i)** | A service is a (background) task provided by the OS that makes useful functions available to all applications, eg GUI, communication between applications, security, file management. | **1 KU** |
| | | **(ii)** | Any acceptable services:<br>eg Clipboard – enables copying of data from one application to paste in another.<br>eg OLE – encapsulates data in one application into an object which can be opened in another. | **1 PS** |
| | | **(iii)** | If each application had to supply its own code for similar functionality, there would be a lot of wasted effort in producing programs/bloated software, etc.<br>**OR**<br>The producer of the OS can supply a specification which will result in applications having the same 'look and feel' resulting in more user satisfaction<br><br>**1 mark for any valid reason** | **1 PS** |
| | **(b)** | **(i)** | The new software may install a new version of a library routine. Other software that uses this routine may then no longer work with this new version.<br><br>**(Allow 'DLL hell' as an answer with explanation)** | **1 PS** |
| | | **(ii)** | eg The API routines are general purpose so may be inefficient for some processes **(1)**. So if the programmer is writing routines that are time critical (for example for real time systems or for complex games), it may be preferable to write the routines specially so that they work as quickly as possible **(1)**.<br><br>**2 marks for any valid reason** | **2 PS** |
| | **(c)** | **(i)** | eg Hidden attribute **(1)**. Files given this attribute cannot be accidentally deleted **(1)**.<br>**OR**<br>eg file size attribute **(1)** can be checked to see if it is correct – file may have been infected by a virus **(1)**<br>**Any valid answer with explanation (2)** | **2 KU** |
| | | **(ii)** | File permissions can be used **(1)**. These give individual users or user groups additional security attributes on files and folders. eg without 'Delete' permission on a file, a user cannot delete a file **(1)**.<br><br>**(or 2 marks for a full valid answer)** | **2 KU** |
| | | **(iii)** | Although the user has 'full control' permissions on the file because it is in his home folder, it has been given a 'Read Only' attribute so cannot be deleted.<br>**OR**<br>File is open/in use. | **1 PS** |

| | | | | Marks |
|---|---|---|---|---|
| | (d) | 1 | **Graphical user interface** detects which icon has been selected and hence the name of the file; displays the hourglass to indicate processing is taking place. | **4 PS** |
| | | 2 | **File Management** determines the location (sector/track numbers) of the file on the hard drive. | |
| | | 3 | **Input/Output system** is used to interact with the file directory to 'delete' the file. | |
| | | 4 | **Memory management** clears the memory used in RAM | |
| | | 5 | **Graphical user interface** deletes the file's icon from the screen if successful; turns off the hourglass. | |
| | | | **1 mark for each, 4 maximum** | |
| **18** | **(a)** | **(i)** | eg Variable partitioning **(1)**. The OS uses an algorithm to find a block of memory large enough to hold the process **(1)**.<br>**OR**<br>eg Best Fit algorithm **(1)**. The whole of the memory is scanned and a block that is just bigger than the size of the process is chosen **(1)**. | **2 KU** |
| | | **(ii)** | Fragmentation will occur – lots of small blocks of memory are available but none large enough to hold a process. | **1 PS** |
| | **(b)** | **(i)** | A queue is most appropriate as it is a first in, first out structure. Processes will be run in the order in which they entered the queue. | **1 PS** |
| | | **(ii)** | A process is at the head of the queue and the current running process ends or is pre-empted **(1)**.<br>A running process becomes blocked by waiting for input/output **(1)**<br>A process terminates **(1)**.<br>An interrupt causes the process to pause **(1)**<br>**2 marks for any valid two answers** | **2 PS** |

[END OF SECTION II − Part B]

**SECTION II – Part C – Computer Networking**

***Marks***

**19. (a)** Accept any **two** of the following: **2 KU**

- To/CC/BCC – e-mail recipient.
- Date – date and time e-mail is sent.
- Received – notes each server which passes on the e-mail.
- Message-ID – unique identifier on originating server
- or any other valid header with appropriate description.

**(b)** Accept any **one** of the following: **1 KU**

- Recipient's computer may not be online.
- Recipient might collect mail from multiple locations
- or any other valid answer.

**(c) (i)** Accept **one** of the following: **1 PS**

- Software can be tailor-made to carry out the tasks required by the multinational company.
- Software can overcome efficiencies of existing standards by only implementing required features, and not including code required only to interoperate with existing standards.

**(ii)** Accept **one** of the following: **1 KU**

- Development team will not have to design a protocol, rather they can follow the standard.
- Following an existing standard could mean that existing module libraries could be used to aid the development of the new software.

**(iii)** The customer could make use of a large number of different, existing, standard-based e-mail clients to connect to their system. **1 KU**

**(d) (i)** Application **1 PS**

**(ii)** **2 PS**

- Application
- Presentation
- Session

**2 marks for all three**
**1 mark for any two**

**(e)** **3 KU**

- Encodes attachment into plain text.
- Adds e-mail headers stating content type of attachment.
- Receiving computer decodes attachment.

***Marks***

**20.** **(a)** Accept **three** of the following: **3 KU**

- Allows varying sizes of network to be allocated, instead of the fixed sizes of class A, B and C networks.
- Less wasted addresses by allocating a class B network (which would waste over 60,000 addresses).
- Routing table entries can be grouped using CIDR.
- Reducing size of routing tables to manageable size.

**(b)** **(i)** Accept **two** of the following statements: **4 PS**

- Less network congestion **(1)** as each building's traffic is limited to each building's subnet **(1)**.
- Increased security **(1)** by restricting access, or containing risks, to individual subnets **(1)**.
- Geographical spread of buildings **(1)** suits administering a subnet for each building **(1)**.
- Easier to rectify faults **(1**) as subnet displaying faults can be isolated **(1)**

**Maximum of two reasons fully explained**

**(ii)** Accept **one** of the following: **2 PS**

255.255.255.128
11111111.11111111.11111111.10000000

**1 mark for the first three octets**
**1 mark for the fourth octet**

**(iii)** Accept **two** of the following: **2 PS**

155.20.42.128
155.20.43.0
155.20.43.128

**1 mark for each subnet, maximum of 2 marks**

| | | | | Marks |
|---|---|---|---|---|
| **21** | **(a)** | | <a href=”http://www.web4u.co.uk”>Web page professionally designed by Web4U.</a> <br><br>**1 mark for centred paragraph (accept use of  tag)**<br>**1 mark for correctly formatted anchor tag**<br>**1 mark for correct syntax throughout** | **3 PS** |
| | **(b)** | **(i)** | Traceroute records which devices are used to route the ICMP packets to their destination. | **1 KU** |
| | | **(ii)** | Accept **two** of the following<br>• Remote host is offline<br>• Router/switch is faulty<br>• Incorrect routing tables<br>• Firewall/proxy restricting access<br>• Loose/faulty cabling. | **2 PS** |
| | **(c)** | **(i)** | • Smurf attack.<br>• Send an ICMP Echo Request to network broadcast address.<br>• Return address is spoofed to be the target host.<br>**OR**<br>• Ping of death.<br>• Sends an oversized ICMP Echo Request.<br>• Vulnerable targets will crash. | **3 PS** |
| | | **(ii)** | • Deny incoming Ping requests.<br>• Deny incoming traffic to the network broadcast messages. | **2 PS** |
| **22** | **(a)** | **(i)** | Accept **two** of the following:<br>• Encodes a signal for transfer over a network.<br>• Synchronise sound and video data.<br>• Transfer of signal over the network.<br>• Decodes received signals. | **2 KU** |
| | | **(ii)** | Accept any of the following:<br>• Greater bandwidth network connection.<br>• Faster network card.<br>• Higher specification video capture hardware. | **2 PS** |
| | **(b)** | **(i)** | Accept **three** of the following:<br>• Sender & recipient create private/public key pairs<br>• Sender requests public key from recipient<br>• File encrypted with recipient’s public key.<br>• Cyphertext can be safely distributed over open channel.<br>• File decrypted with recipient’s private key. | **3 KU** |
| | | **(ii)** | No need to find secure channel to communicate shared key. | **1 PS** |

| | | | | Marks |
|---|---|---|---|---|
| **23** | **(a)** | • Full back up once a week, overnight (must emphasise timing)<br>• Incremental back-ups twice a day, morning and afternoon.<br>• Differential back-ups each evening after stores close<br><br>**or equivalent distribution of back-up types.** | | **3 PS** |
| | **(b)** | **(i)** | Accept **two** of the following:<br>• Tunnelling protocol establishes connection to remote network.<br>• Networking traffic encapsulated within tunnelling protocol.<br>• Client uses features of remote network as if physically connected. | **2 KU** |
| | | **(ii)** | L2TP (Layer 2 Tunnelling Protocol). | **1 KU** |
| | | **(iii)** | Accept **two** of the following:<br>• Built-in encryption, rather than relying on encapsulated protocol.<br>• Supports more than a single point-to-point tunnel.<br>• Can be used on non-IP networks. | **2 PS** |
| **24** | **(a)** | Accept any of the following:<br>• High bandwidth, 10 Gbps **(1)** – shared traffic from whole buildings **(1)**.<br>• Long range, 3 km **(1)** – enough to reach across town **(1)**.<br>• Security **(1)** difficult to eavesdrop without breaking connection **(1)**.<br><br>**Maximum of two reasons fully explained** | | **4 PS** |
| | **(b)** | **(i)** | 802.11g, 802.11n, HiperLAN 2<br><br>**Do not accept** 802.11a/b due to limited transfer rate and range. | **1 KU** |
| | | **(ii)** | • MAC address filtering to only allow known devices access.<br>• WEP/WPA encryption to prevent data interception.<br>• SSID and network key to prevent unauthorised access. | **3 PS** |
| | **(c)** | • Upgraded network provision given to small selection of employees.<br>• Employees give performance feedback to engineers. | | **2 PS** |

[END OF SECTION II – Part C]

[END OF MARKING INSTRUCTIONS]