A-level
Computer Science

Paper 1

Mark scheme

Mark schemes are prepared by the Lead Assessment Writer and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation events which all associates participate in and is the scheme which was used by them in this examination. The standardisation process ensures that the mark scheme covers the students' responses to questions and that every associate understands and applies it in the same correct way. As preparation for standardisation each associate analyses a number of students' scripts: alternative answers not already covered by the mark scheme are discussed and legislated for. If, after the standardisation process, associates encounter unusual answers which have not been raised they are required to refer these to the Lead Assessment Writer.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of students' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Further copies of this Mark Scheme are available from http://www.aqa.org.uk/

COMPONENT NUMBER: **Paper 1**

COMPONENT NAME:

STATUS:

DATE: **20 February 2015**

The following annotation is used in the mark scheme.

;	- means a single mark
//	- means alternative response
/	- means an alternative word or sub-phrase
A	- means acceptable creditworthy answer
R	- means reject answer as not creditworthy
NE	- means not enough
I	- means ignore
DPT	- in some questions a specific error made by a candidate, if repeated, could result in the failure to gain more than one mark. The **DPT** label indicates that this mistake should only result in a candidate losing one mark, on the first occasion that the error is made. Provided that the answer remains understandable, subsequent marks should be awarded as if the error was not being repeated.

Level of response marking instructions.

Level of response mark schemes are broken down into a number of levels, each of which has a descriptor. The descriptor for the level shows the average performance for the level. There are a range of marks in each level. The descriptor for the level represents a typical mid-mark performance in that level.

Before applying the mark scheme to a student's answer read through the answer and annotate it (as instructed) to show the qualities that are being looked for. You can then apply the mark scheme.

Step 1 Determine a level

Start at the lowest level of the mark scheme and use it as a ladder to see whether the answer meets the descriptor for that level. The descriptor for the level indicates the different qualities that might be seen in the student's answer for that level. If it meets the lowest level then go to the next one and decide if it meets this level, and so on, until you have a match between the level descriptor and the answer. With practice and familiarity you will find that for better answers you will be able to quickly skip through the lower levels of the mark scheme.

When assigning a level you should look at the overall quality of the answer and not look to pick holes in small and specific parts of the answer where the student has not performed quite as well as the rest. If the answer covers different aspects of different levels of the mark scheme you should use a best fit approach for defining the level and then use the variability of the response to help decide the mark within the level. ie if the response is predominantly level 3 with a small amount of level 4 material it would be placed in level 3 but be awarded a mark near the top of the level because of the level 4 content.

Step 2 Determine a mark

Once you have assigned a level you need to decide on the mark. The exemplar materials used during standardisation will help. There will be an answer in the standardising materials which will correspond with each level of the mark scheme. This answer will have been awarded a mark by the Lead Examiner. You can compare the student's answer with the example to determine if it is the same standard, better or worse than the example. You can then use this to allocate a mark for the answer based on the Lead Examiner's mark on the example.

You may well need to read back through the answer as you apply the mark scheme to clarify points and assure yourself that the level and the mark are appropriate.

Indicative content in the mark scheme is provided as a guide for examiners. It is not intended to be exhaustive and you must credit other valid points. Students do not have to cover all of the points mentioned in the indicative content to reach the highest level of the mark scheme.

An answer which contains nothing of relevance to the question must be awarded no marks.

Qu	Part	Marking guidance	Total marks
01	1	**Mark is for AO1 (knowledge)** **1 mark**: B;	1
01	2	**Mark is for AO1 (knowledge)** **1 mark**: A;	1
01	3	**Mark is for AO1 (understanding)** It demonstrates that there are some (well-defined) problems that cannot be solved by a computer;	1
02	1	**Mark is for AO1 (knowledge)** A subroutine that calls itself;	1

02	2	**Mark is for AO2 (analyse)** When S is greater than or equal to E;	1

02	3	**All marks AO2 (apply)**	6

Call number	S	E	M	List returned
1	1	5	3	
2	1	3	2	
3	1	2	1	
4	1	1		[6]
3	1	2	1	
5	2	2		[3]
3	1	2	1	[6, 3]
2	1	3	2	
6	3	3		[4]
2	1	3	2	[6, 4, 3]
1	1	5	3	
7	4	5	4	
8	4	4		[8]
7	4	5	4	
9	5	5		[5]
7	4	5	4	[8, 5]
1	1	5	3	[8, 6, 5, 4, 3]

Mark as follows:
1 mark: Correct list returned by call number 3;
1 mark: Correct lists returned by call number 6-9;
1 mark: Correct list returned by call number 1;
1 mark: S, M, E given correct values for call number 6;
1 mark: S, M, E given correct values for call number 7;
1 mark: S, M, E given correct values for call number 8-9;

Info for examiner: Ignore missing values for S, E, M after the first time a particular call number appears in the table.

02	4	**Mark is for AO2 (analyse)** nlog n **A.** O(nlog n) **A.** O(nlog$_2$n) **A.** nlog$_2$n **NE.** log(n)	1

02	5	**All marks AO1 (knowledge)** Return address; Parameters; Local variables; Return value; **MAX 2**	2

02	6	**All marks AO2 (analyse)** Because there will be four recursive calls to MergeSort (1, 2, 3, 4) and (1, 2, 3, 5); there will also be three recursive calls to MergeSort (1, 2, 3) and one call to Merge.	2

03	1	**All marks AO1 (understanding)** Easier to test each class using unit testing // with composition each class can be tested separately but it is not possible to test a subclass independently from the base class; There can be unintended side-effects for derived classes if a method in the base class is altered; Composition is more flexible as if a new class is developed it can easily be used instead of the class that currently is used in the composition; **MAX 1**	1

03	2	**All marks AO2 (apply)**	4

```
Buyer = Class(Client) {
Private:
  NoOfBedroomsRequired: Integer
  OffStreetParking: Boolean
  AreaDesired: String
Public:
  Function GetNoOfBedroomsRequired
  Function GetOffStreetParking
  Function GetAreaDesired
  Procedure SetDetails (Override)
}
```

Info for examiner: Accept answers that use different notations, so long as meaning is clear.

Mark as follows:

1 mark for correct header including name of class and parent class;
1 mark for redefining the SetDetails procedure;
1 mark* for defining all 3 extra functions needed to read variable values, all identified as being public (keyword public is optional if functions are declared before variables);
1 mark** for defining all 3 extra variables, with appropriate data types and identified as being private;
A. answers that indicate separately that each variable is private or each method is public
R. do not award mark for declaring new functions if any of the functions have the same name as the variables
I. parameters to methods, minor changes to names that do not affect clarity

| | | * - Do not award this mark if any extra functions/procedures have been declared, except for functions that would set values e.g. `SetAreaDesired` an incorrectly named procedure to add e.g. `AddBuyer`

** -Do not award this mark if any extra variables have been declared | |

03	3	**All marks AO1 (knowledge)** Composition and aggregation are both "has a" relationships – when an object contains another object; With composition if the containing object is destroyed so are the objects in contains, this is not the case with aggregation;	2

03	4	**One mark is for AO1 (understanding) and one mark is for AO2 (apply)** **1 mark for AO2:** the Client class contains an object of the class Location; **1 mark for AO1:** If composition had been used then if the client object is destroyed so is the address object – but in real life the address object would still exist;	2

04	1	**All marks AO1 (knowledge)** A set is an unordered collection (of values); in which each value occurs at most once // with no duplicates;	2

04	2	**Mark is for AO2 (apply)** S3 // S5;	1

04	3	**Mark is for AO2 (apply)** S5;	1

04	4	**Mark is for AO2 (apply)** The difference between set S5 and S1 // the difference between set S2 and S1 // S5 – S1 // S2 – S1;;	1

04	5	**All marks AO2 (apply)** {(a, c); (b, c);} // {(c, a); (b, c);} // {(x, y) : x ←S1, y←S6};; ,	2

		MAX 1 if additional pairs listed	

04	6	**All marks AO2 (apply)** a | ab; ab?; **A.** any correct answer	2

05	1	**Mark is for AO2 (apply)** Katherine Hepburn;	1

05	2	**Mark is for AO2 (apply)** 52;	1

05	3	**All marks AO1 (knowledge)** Hashing algorithm is applied to the key; The result is an index of where to store the value in an array; It is possible that the hashing algorithm might generate the same key for two different values (this is called a collision); in which case a collision handling method (**A.** example of collision handling method) is used;	4

05	4	**All marks AO1 (knowledge)** A larger hash table is created; Each value in the existing table is inserted into the new table; In a position determined by a new hashing algorithm; **A.** In a position determined by the same algorithm but with larger table size;	3

05	5	**Mark for AO1 (understanding)** By making the value be a data structure (**A.** any suitable example of data structure) that allows more than one catalogue number to be stored (this is necessary because it is not possible to have multiple instances of a key in a dictionary)	1