CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Advanced Level

MARK SCHEME for the May/June 2015 series

9691 COMPUTING

9691/33 Paper 3 (Written Paper), maximum raw mark 90

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2015 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

CAMBRIDGE
International Examinations

1 (a) (i) The table has a repeated (group of) attributes **[1]**

(ii) `Title, Genre, ReleaseDate` and `ReviewDate` are repeated for each reviewer **[1]**

(b) (i)

ReviewerID	Location
510	London
808	New York
756	Dhaka

[1]

(ii)

Title	Genre	ReleaseDate	ReviewDate	ReviewerID
Hits 36	P	12/01/2015	01-15	510
Popular Bach	C	12/01/2015	02-15	808
The Messiah	C	11/1/2014	11-14	756

Or, any other row taken from the original table (`ReviewID` must be different) **[2]**

3 correct – 2 marks
2 correct – 1 mark
1 correct only – scores 0

(iii) 9 **[1]**

(iv) Many-to-one **[1]**

(v) Primary key/`ReviewerID` in the `REVIEWER` table **[1]**

Links to foreign key/`ReviewerID` in the `REVIEW` table **[1]**

(c) (i) `Title` **[1]**

(ii) There are non-key attributes which are dependent (may be stated as part of the attribute description) (1)

`ReviewerName` is dependent on `ReviewerID` //
`Fee` is dependent on `Genre` (1) **[2]**

(iii) `REVIEWER(ReviewerID, Location, ReviewerName)`
`REVIEW(Title, Genre, ReleaseDate,`
`ReviewDate, ReviewerID)`
`FEE(Genre, GenreFee)`

Mark as follows:
new `FEE` table
containing `Genre` and `GenreFee`
Primary key for `FEE` correctly shown

`REVIEW` table has foreign key `Genre`
`REVIEWER` table contains `ReviewerName` **[5]**

[Total: 17]

2 (a) Syntax diagram **[1]**

(b) (i) The rule is defined in terms of itself / calls itself **[1]**

(ii) TRUE
FALSE
FALSE

CAO. **[1]**

(c) (i) `D175N`

Invalid (1)
5, 2 (1) **[2]**

(ii) `W058M`

Valid (1)

Rule 1 – 3 times
Rule 2 – once
Rule 3 – once
Rule 4 – once
Rule 5 – once (1) **[2]**

(iii) `C86N`

Invalid (1)
`<producttype><digit><digit><location>`

Rule 1 – 2/3 times
Rule 2 – once
Rule 4 – once
Rule 5 – once (1) **[2]**

[Total: 9]

3 (a) *The class diagram includes:*

`Software + Network` subclasses (1)

`Bespoke + OffTheShelf` subclasses of `Software` (1)

Note: Two marks – correct class names only

Recognised notation for inheritance (from `Software` and `Network` only) (1)

`Project` class
```
StartDate     : DATE
ProjectLeader : STRING
```
(1)

`Software` class
```
ProgrammingLanguage : STRING
AlphaTesting        : STRING/CHAR
```
(1)

For each of the following – <u>at least two</u> of the correct properties

`Bespoke` class
```
CustomerName        : STRING
AgreedCost          : REAL/CURRENCY
AgreedDeliveryDate  : DATE
```
(1)

`OffTheShelf` class
```
Title         : STRING
BetaTesting   : STRING/CHAR
RetailPrice   : REAL/CURRENCY
SalesForecast : INTEGER
```
(1)

`Network` class
```
ClientName    : STRING
AgreedCost    : INTEGER/CURRENCY
```
(1) **[8]**

(b) (i) *Class…*
The definition of an object // The 'blueprint' from which objects are created. **[1]**

(ii) *Inheritance…*
The ability of a subclass/child class to use properties and methods of a parent/super/base class **[1]**

(c) (i) *Instance …*
– A specific object created from a class (1)

– Main program has created an instance of the `Network` class – referred to as `ThisNetworkProject` (1) **[2]**

(ii) *Method…*
Something an object can do // Implemented with procedures/functions (1)

`Project` shows two methods – `get_ProjectID()` and `set_ProjectID()` (1) **[2]**

(iii) *Encapsulation…*
Technique which restricts the programmer's access to the object's data (1)
Data values can only be read/written using methods provided by the class (1)

– The `ProjectID` value can only be read/written using the two methods provided
`// ProjectID` is private to the class (1) **[3]**

[Total: 17]

4 **(a)** Last item in is the first item out // First item in is the last item out
R. LIFO **[1]**

(b)
```
PROCEDURE InitialiseStack
    FOR Index ← 0 to 99
        Animal[Index] ← ""                    (1)
    ENDFOR
    StackPointer ← -1                         (1)
ENDPROCEDURE                                  [2]
```

(c) **(i)** "" //empty string (1)
1 (1) **[2]**

(ii)
```
PROCEDURE Push
    IF StackPointer = 99                      (1)
        THEN
            OUTPUT "REFUSED – stack is full"
        ELSE
            INPUT NewAnimal                   (1)
                StackPointer ← StackPointer + 1   (1)
                Animal[StackPointer] ← NewAnimal  (1)
    ENDIF
ENDPROCEDURE                                  [4]
```

(d)
```
PROCEDURE Pop
    IF StackPointer = -1
        THEN
            OUTPUT "Stack is empty"
        ELSE
            OUTPUT Animal[StackPointer]
            StackPointer ← StackPointer – 1
    ENDIF
ENDPROCEDURE
```

Mark as follows …
Test for empty with `StackPointer = -1` **[1]**
`OUTPUT 'EMPTY' message` **[1]**
`Animal[StackPointer]` is value removed **[1]**
Decrement `StackPointer` **[1]**

[Total: 13]

5 (a) (i) –23 (1)

E9 (1) **[2]**

(ii) 107 (1)

6B (1) **[2]**

(iii) 127 **[1]**

(iv) Fewer digits used to represent any number // long string difficult to interpret (1)

Less likely to make a mistake when copying/converting a digit string (1)

Easy to convert from binary/denary to hex (vice versa) (than binary to denary) (1)

[Max 1]

(b)

99	0	1	1	0	0	0	1	1
29	0	0	0	1	1	1	0	1
	1	0	0	0	0	0	0	0

+

99 and 29 correct pattern (1)

Correct addition // ft (1)

Overflow has occurred // the expected answer is outside the possible range // the answer is showing as –128 (1) **[3]**

(c) (i) `1578` **[1]**

(ii) `1101` is not a valid BCD digit string // `1101` represents '13' **[1]**

[Total: 11]

6 (a) Systems flowchart **[1]**

(b)
- 1 – Text editor
- 2 – on-screen errors
- 3 – Compiler
- 4 – Assembler
- 5 – Linker
- 6 – Program library code
- 7 – Executable code **[7]**

[Total: 8]

7 **(a)** *Correct use of any of the following:*

Single segment of wire (1)
Computers connected to the cable X 4 (1)
Terminators X 2 (1)
Computer C has attached printer (1)
File server (1)
Firewall / Proxy server + Indication of a connection to the Internet (1)
Router + Indication of a connection to the Internet (1)
Modem + Indication of a connection to the Internet (1)

[Max 5]

(b) Manage user accounts (1)
Authenticate all logons (1)
Manage the shared file storage (1)
Manage the installation and use of applications software (1)

[Max 3]

(c) *Intranet …*

Information system using Internet protocols (1)
Provides service of web pages (to client computers) (1)
Information system only available to staff within the organisation (1)

[Max 2]

[Total: 10]

8 **(a)** **(i)** `ChangeString` (1)
`ThisString1, ThisString2` (only) (1) **[2]**

(ii) `Ali J` **[1]**

(iii) 7 // Error if LEFT("", 1) generates an error **[1]**

(iv) `JONES 8` **[1]**

[Total: 5]