MARK SCHEME for the October/November 2012 series

9691 COMPUTING

9691/22 Paper 2 (Written Paper), maximum raw mark 75

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 October/November 2012 series for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level components and some Ordinary Level components.
1 (a)

Super Cars

Enter car details
Enter car hire details

Enter hirer
Enter car
Enter payment

1 mark per row in correct order [2]

(b) – to enable modular testing/maintenance/debugging
– to enable different blocks to be worked on by different staff
– easier to understand // reduce complexity [2]

(c)

Super Cars

Enter car details
Enter car hire details

Input car’s identification details
Input hire rates
Enter hirer
Enter car
Enter payment

1 mark for 2 blocks under Enter Car Details [1]

(d) (i) Invalid

(ii) Invalid

(iii) grey is valid [3]
Delphi Pascal

VAR CarRegValid : BOOLEAN;
  CarReg : STRING;
BEGIN
  CarRegValid := TRUE;
  READLN(CarReg);
  IF LENGTH(CarReg) <> 6 THEN
    CarRegValid := FALSE;
  IF NOT((COPY(CarReg, 1, 2) >= '00') AND (COPY(CarReg, 1, 2) <= '99')) THEN
    CarRegValid := FALSE;
  IF COPY(CarReg, 3, 4) <> 'HIRE' THEN
    CarRegValid := FALSE;
  IF CarRegValid THEN
    WRITELN('Valid')
  ELSE
    WRITELN('Invalid');
END.

VB 2005

Dim CarRegValid As Boolean
Dim CarReg As String
CarRegValid = True
CarReg = Console.ReadLine()
If Len(CarReg) <> 6 Then
  CarRegValid = False
End If
If Not (Mid(CarReg, 1, 2) >= "00" And Mid(CarReg, 1, 2) <= "99") Then
  CarRegValid = False
End If
If Mid(CarReg, 3, 4) <> "HIRE" Then
  CarRegValid = False
End If
If CarRegValid Then
  Console.WriteLine("Valid")
Else
  Console.WriteLine("Invalid")
End If

VB6

Dim CarRegValid As Boolean
Dim CarReg As String
CarRegValid = True
CarReg = InputBox"
If Len(CarReg) <> 6 Then
  CarRegValid = False
End If
If Not (Mid(CarReg, 1, 2) >= "00" And Mid(CarReg, 1, 2) <= "99") Then
  CarRegValid = False
End If
If Mid(CarReg, 3, 4) <> "HIRE" Then
  CarRegValid = False
End If
If CarRegValid Then
  MsgBox("Valid")
Else
  MsgBox("Invalid")
End If
Python

carReg = input()
carRegValid = True
if len(carReg) != 6 :
    carRegValid = False
if not(carReg[0 : 2] >= '00' and carReg[0 : 2] <= '99') :
    carRegValid = False
if carReg[2 : 6] != 'HIRE' :
    carRegValid = False
if carRegValid :
    print("Valid")
else :
    print("Invalid")

1 mark for length check (accept incomplete check)
1 mark for correct separating 1st two characters
1 mark for testing first two characters are digits
1 mark for separating last four characters
1 mark for testing last four characters are HIRE
1 mark for initialising Boolean value
1 mark for changing Boolean value if error
1 mark for suitable message for valid and invalid
1 mark for correct use of specified programming language
1 mark for indentation

(ii) – string length > 6 // three leading digits (instead of 2)

– Line number quoted must include the condition

(f) (i) Alpha testing:
Who – issue of software to a restricted number of testers within the company
When – it may not be completely finished and could have faults // before beta testing
Purpose – to find faults // to check the logic // to see if it works

(ii) Beta testing:
Who – released to specific customers // potential users
When – in finished state // after alpha testing // before release of software
Purpose – for their constructive comments // feedback // to find errors missed earlier
2 (a)

<table>
<thead>
<tr>
<th>Row</th>
<th>Position</th>
<th>Row&lt;=$25</th>
<th>Position&lt;=$4</th>
<th>CarReg&lt;&gt;&quot;00HIRE&quot;</th>
<th>ParkingSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>52HIRE</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td>10HIRE</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td>67HIRE</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td>24HIRE</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>FALSE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>TRUE</td>
<td></td>
<td></td>
<td>63HIRE</td>
</tr>
</tbody>
</table>

1 mark for second decision in heading
1 mark for third decision in heading
1 mark for correct array elements in heading
1 mark for correct values into array elements
1 mark for correct values in column 2
1 mark for correct placing of the FALSE

(b) Pascal

```pascal
Row := 1;
WHILE Row <= 25 DO
BEGIN
Position := 1;
WHILE Position <= 4 DO
BEGIN
READLN(CarReg);
IF CarReg = '00HIRE' THEN Exit;
ParkingSpace[Row, Position] := CarReg;
Position := Position +1;
END;
Row := Row + 1;
END;
```

(b) VB 2005

```vbnet
Row = 1
Do While Row <=25
  Position = 1
  Do While Position <= 4
    CarReg = Console.ReadLine()
    If CarReg = '00HIRE' Then Exit Sub
    ParkingSpace(Row, Position) = CarReg;
    Position = Position + 1
  LOOP
Row = Row + 1
LOOP
```
VB6

Row = 1
Do While Row <= 25
    Position = 1
    Do While Position <= 4
        CarReg = InputBox(""")
        If CarReg = "00HIRE" Then Exit Sub
        ParkingSpace(Row, Position) = CarReg
        Position = Position + 1
    Loop
    Row = Row + 1
Loop

Python

Row = 1
while Row <= 25 :
    Position = 1
    while Position <= 4 :
        CarReg = input()
        if CarReg == "00HIRE" :
            return
        ParkingSpace[Row][Position] = CarReg
        Position = Position + 1
    Row = Row + 1

1 mark for correct WHILE loops
1 mark for correctly nested loops (must indicate end of loops)
1 mark for input in correct place
1 mark for correct incrementation (Row and Position)
1 mark for checking for rogue value
1 mark for assignment to correct array element
1 mark for indentation

Check that WHILE, IF and assignment statements are properly formed depending on the programming language [7]

(c) (i) 0 (zero)  [1]
          (Correct answer only)

(ii) Run-time error  [1]

(iii) Check the value of the bracket before the division takes place // write error trapping code
     if bracket = 0 arrange for a message to be output // exception code  [2]
(d) – set breakpoint at the beginning of the code under scrutiny
– at the point(s) in the program where variable values are to be checked
– program runs normally until breakpoint reached
– check for current variable values
– a line/statement/instruction at a time

3  (a)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Size of Field (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CarReg</td>
<td>String/alphanumeric/text</td>
<td>6</td>
</tr>
<tr>
<td>Make</td>
<td>String/alphanumeric/text</td>
<td>10-20</td>
</tr>
<tr>
<td>DateBought</td>
<td>Date/integer/real/string</td>
<td>8</td>
</tr>
<tr>
<td>OnHire</td>
<td>Boolean</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) \((6 + 20 + 8 + 1)\)
\(* \frac{100}{1024}\)
\(* 1.1 \) (or equivalent/similar)
= 3.8 KB
1 mark per row above

(c) (i) Pascal

```pascal
TYPE HireCar = RECORD
   CarReg : String[6];
   Make : String[20];
   DateBought : TDateTime;
   OnHire : Boolean;
END;
```

VB 2005

```vb
STRUCTURE HireCar
   DIM CarReg AS String
   DIM Make AS String
   DIM DateBought AS Date
   DIM OnHire AS Boolean
END STRUCTURE
```

VB6

```vb
Type HireCar
   CarReg As String
   Make As String
   DateBought As Date
   OnHire As Boolean
End Type
```
Python

class HireCar :
    def __init__(self, carReg, make, dateBought, onHire) :
        self.CarReg = carReg
        self.Make = make
        self.DateBought = dateBought
        self.OnHire = onHire

1 mark for correct record structure heading
1 mark for correct record structure ending
1 mark for 2 STRING fields
1 mark for Date field
1 mark for Boolean field

Check programming examples
Penalise once for a repeat mistake

(ii) Pascal

PROCEDURE AddCar(VAR CarRecord);
BEGIN
    AssignFile(CarFile, 'SuperCars');
    Reset(CarFile);
    Seek(CarFile, FileSize(CarFile));
    Write(CarFile, CarRecord);
    CloseFile(CarFile);
END;

VB 2005

SUB AddCar(BYREF CarRecord AS HireCar)
    Writer = New BinaryWriter(CarFile)
    CarFile.Write(CarRecord)
    CarFile.Close()
END SUB

Python

import pickle

def addCar(CarRecord) :
    CarFile = open("SuperCars", "ab")
    pickle.dump(CarRecord, CarFile)
    CarFile.close()

Accept pseudocode

1 mark for correct procedure heading
1 mark for parameter in procedure heading
1 mark for opening file for writing/appending
1 mark for accessing end of file
1 mark for writing record
1 mark for closing file

© Cambridge International Examinations 2012
(d) By value:
- a local copy of the data is used
- leaving the variable in the main program unaffected

By reference:
- the address of the memory location of the data to be used is passed
- so value changes in procedure are also reflected in main program [4]

4
- date (month alone sufficient)
- suitable report title
- the company name (Super Cars)
- tabulated or other suitable layout
- headings/labels (must contain income, car, number of times hired)
- well spaced out (making use of whole frame)
(if clearly a screen design do not give this mark) [6]