

**CAMBRIDGE INTERNATIONAL EXAMINATIONS**  
GCE Advanced Subsidiary Level and GCE Advanced Level

## **MARK SCHEME for the October/November 2012 series**

### **9691 COMPUTING**

**9691/22**

Paper 2 (Written Paper), maximum raw mark 75

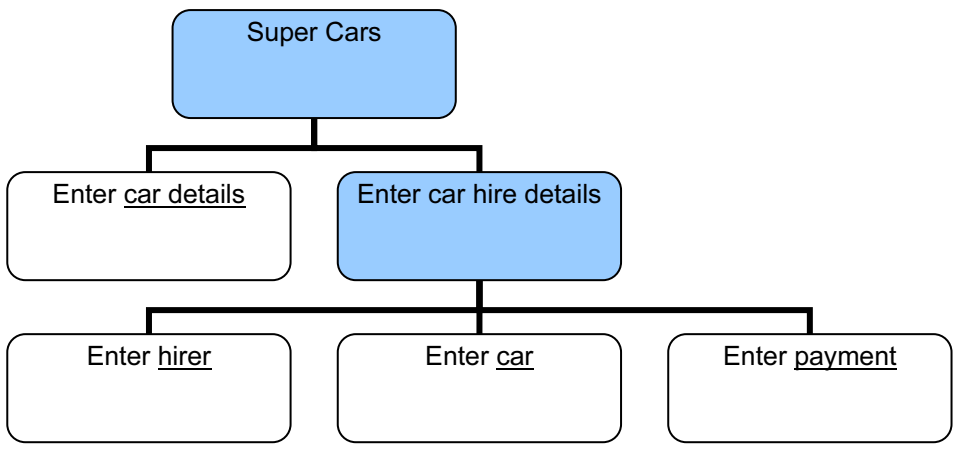
This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began,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

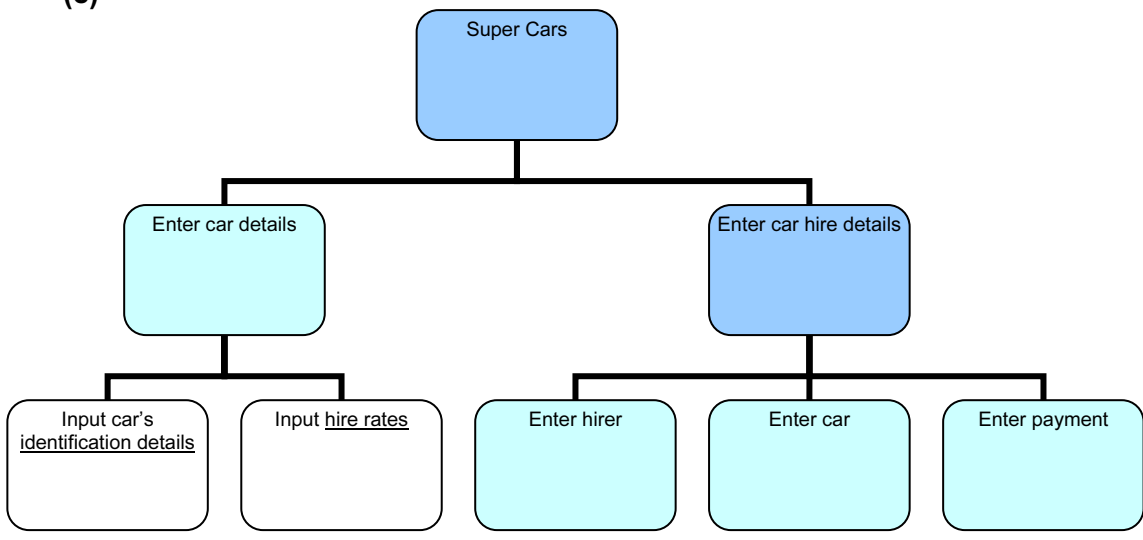
1 (a)



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)



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

(d) (i) Invalid  
 (ii) Invalid  
 (iii) grey is valid [3]

<b>Page 3</b>	<b>Mark Scheme</b>	<b>Syllabus</b>	<b>Paper</b>
	<b>GCE AS/A LEVEL – October/November 2012</b>	<b>9691</b>	<b>22</b>

**(e) (i) 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

```

Page 4	Mark Scheme	Syllabus	Paper
	GCE AS/A LEVEL – October/November 2012	9691	22

## 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 1<sup>st</sup> 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*

[10]

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

– Line number quoted must include the condition

[2]

(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

[3]

(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 [3]

2 (a)

Row	Position	Row<= 25	Position <=4	CarReg <>"00HIRE"	ParkingSpace				
					[1,1]	[1,2]	[1,3]	[1,4]	[2,1]
1	1	TRUE	TRUE	TRUE	52HIRE				
	2		TRUE			10HIRE			
	3		TRUE				67HIRE		
	4		TRUE					24HIRE	
	5		FALSE						
2	1		TRUE						63HIRE

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

[6]

(b) 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;

```

VB 2005

```

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

```

Page 6	Mark Scheme	Syllabus	Paper
	GCE AS/A LEVEL – October/November 2012	9691	22

## 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]

<b>Page 7</b>	<b>Mark Scheme</b>	<b>Syllabus</b>	<b>Paper</b>
	<b>GCE AS/A LEVEL – October/November 2012</b>	<b>9691</b>	<b>22</b>

- (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

[4]

3 (a)

Field Name	Data Type	Size of Field (bytes)
CarReg	String/alphanumeric/text	6
Make	String/alphanumeric/text	10-20
DateBought	Date/integer/real/string	8
OnHire	Boolean	1

[4]

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

[4]

(c) (i) **Pascal**

```

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

**VB 2005**

```

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

**VB6**

```

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

<b>Page 8</b>	<b>Mark Scheme</b>	<b>Syllabus</b>	<b>Paper</b>
	<b>GCE AS/A LEVEL – October/November 2012</b>	<b>9691</b>	<b>22</b>

### 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*

[5]

### (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)
    CarFile = New FileStream ('SuperCars', FileMode.Append)
    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*

[5]



<b>Page 9</b>	<b>Mark Scheme</b>	<b>Syllabus</b>	<b>Paper</b>
	<b>GCE AS/A LEVEL – October/November 2012</b>	<b>9691</b>	<b>22</b>

- (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]