CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Advanced Level

MARK SCHEME for the May/June 2015 series

9608 COMPUTER SCIENCE

9608/43 Paper 4 (Written Paper), maximum raw mark 75

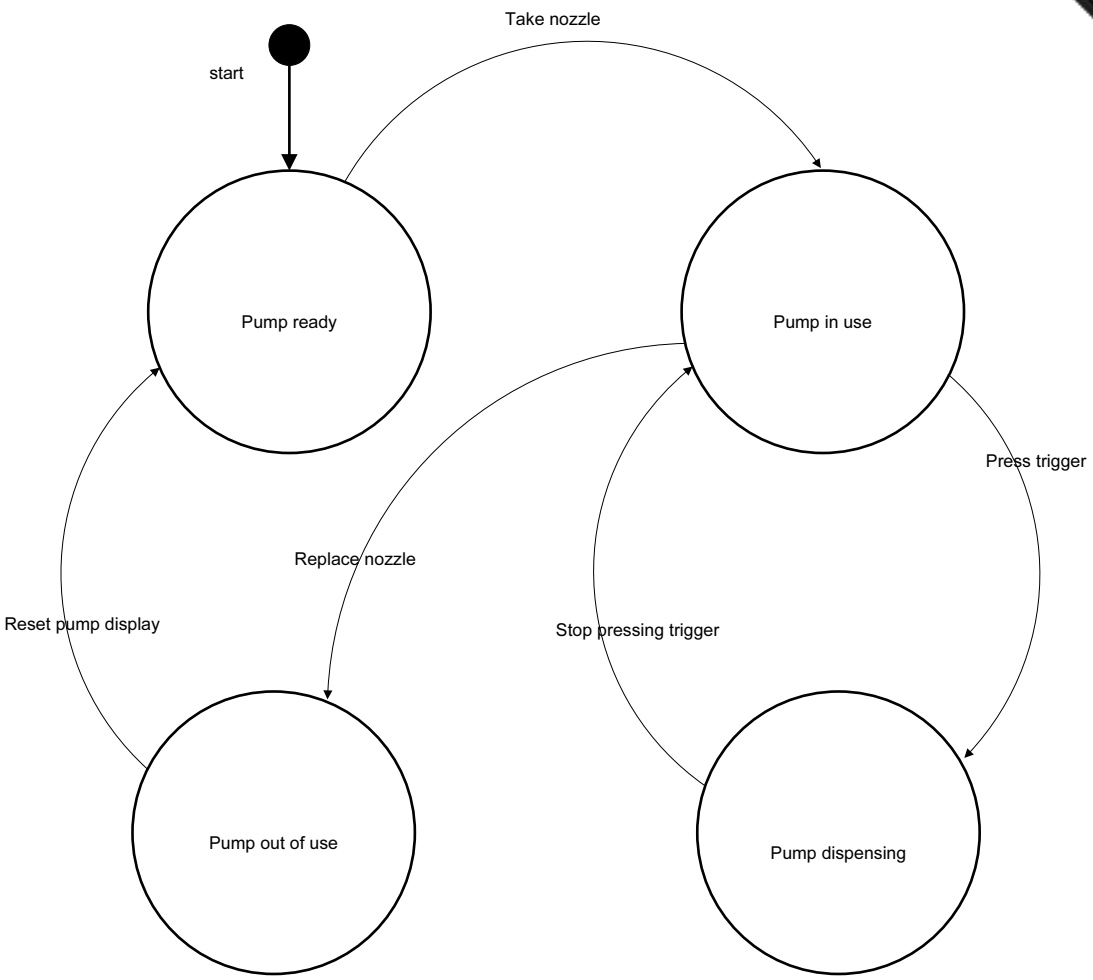
This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, 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 May/June 2015 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

1



[9]

2 **(a)**
```
made_with(laasi, milk).
made_with(laasi, yogurt).
dairy_product(milk).
dairy_product(yogurt).
```
[4]

(b) Ingredient =
cheese, egg, flour **[2]**

(c) contains_meat(Dish)
IF
made_with(Dish, X) (2 marks)
AND (1 mark)
meat(X) (1 mark) **[4]**

3 (a)

Conditions	Age under 25	Y	Y	Y	Y	N	N	N	N
	Previous accident	Y	Y	N	N	Y	Y	N	N
	Licence held for 3 or more years	Y	N	Y	N	Y	N	Y	N
Actions	10% extra cost		X						
	No discount	X			X	X	X		
	5% discount			X				X	X
		1 mark	1 mark	1 mark	1 mark	1 mark		1 mark	

[6]

(b)

Conditions	Age under 25	Y	Y	Y	Y	N	N		
	Previous accident	Y	Y	N	N	Y	N		
	Licence held for 3 or more years	Y	N	Y	N	-	-		
Actions	10% extra cost		X						
	No discount	X			X	X			
	5% discount			X			X		
		1 mark				1 mark	1 mark		

[3]

(c) Example Pascal

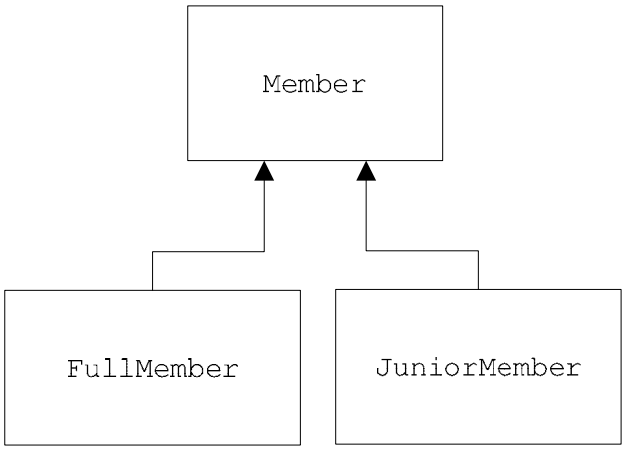
```
FUNCTION CostPercentageChange(DriverAge : INTEGER;
      HadAccident : BOOLEAN; YearsLicenceHeld : INTEGER) : INTEGER;
  BEGIN
      IF DriverAge >= 25
        THEN
          IF HadAccident = TRUE
            THEN
              CostPercentageChange := 0
            ELSE
              CostPercentageChange := -5
        ELSE
          IF HadAccident = TRUE
            THEN
              IF YearsLicenceHeld < 3
                THEN
                  CostPercentageChange := 10
                ELSE
                  CostPercentageChange := 0
            ELSE
              IF YearsLicenceHeld < 3
                THEN
                  CostPercentageChange := 0
                ELSE
                  CostPercentageChange:= -5;
  END;
```

Example Python

```
def CostPercentageChange(DriverAge, HadAccident, YearsLicenceHeld) :
  if DriverAge >= 25:
    if HadAccident:
      return 0
    else:
      return -5
  else:
    if HadAccident:
      if YearsLicenceHeld < 3:
        return 10
      else:
        return 0
    else:
      if YearsLicenceHeld < 3:
        return 0
      else:
        return -5;
```

Mark as follows:
Correct function header
Correct IF statement (1)
Correct IF statement (2)
Correct IF statement (3)
Correct IF statement (4)
Correct IF statement (5)
Correct return statement (or equivalent) **[max 6]**
OR equivalent demonstrating correct logic

4 (a)



[3]

(b) Example Pascal

```
Member = CLASS
         PUBLIC
           Procedure SetMemberName;
           Procedure SetMemberID;
           Procedure SetSubscriptionPaid;
         PRIVATE
           MemberName       : STRING;
           MemberID         : STRING;
           SubscriptionPaid : Boolean;
   END;
```

Example Python

```
class Member() :
    def__init__(self):          PUBLIC
        self.__MemberName = ""
        self.__MemberID = ""
        self.__SubscriptionPaid = False
    def SetMemberName(self, Name):
        self.MemberName = Name
    def SetMemberID(self, ID):
        self.MemberID = ID
    def SetSubscriptionPaid(self, Paid):
        self.SubscriptioPaid = Paid
```

Mark as follows:

Class header	(1 mark)	
`Public` and `Private` used correctly	(1 mark)	
`MemberName` + `MemberID`	(1 mark)	
`SubscriptionPaid`	(1 mark)	
Methods × 3	(1 mark)	**[5]**

(c) (i) Example Pascal

```
JuniorMember = CLASS (Member)
                  PUBLIC
                    Procedure SetDateOfBirth;
                  PRIVATE
                    DateOfBirth : DateTime;
               END;
```

Example Python

```
class JuniorMember (Member):
    def__init__self:
        super().__init__()
        self.DateOfBirth = ""
    def SetDateOfBirth(self, Date):
        self.DateOfBirth = Date
    def SetMemberName(self, Name):
        super().SetMemberName(Name)
    def SetMemberID(self, ID):
        super().SetMemberID(ID)
    def SetSubscriptionPaid(self, Paid):
        super().SetSubscriptioPaid(Paid)
```

[3]

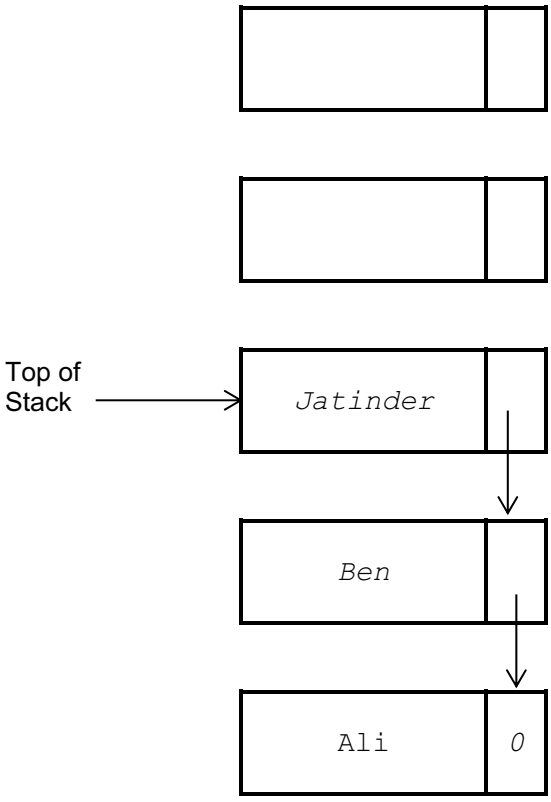
(ii) Example Pascal

```
NewMember := JuniorMember.Create;                    (1 mark)
NewMember.SetMemberName('Ahmed');
NewMember.SetMemberID('12347');                      (1 mark)
NewMember.SetSubscriptionPaid(TRUE);
NewMember.SetDateOfBirth("12/11/2001");              (1 mark)
```

Example Python

```
NewMember := JuniorMember()                          (1 mark)
NewMember.SetMemberName("Ahmed")
NewMember.SetMemberID("12347")                       (1 mark)
NewMember.SetSubscriptionPaid(TRUE)
NewMember.SetDateOfBirth("12/11/2001")               (1 mark)   [3]
```

5 (a)



1 mark for Top of Stack pointer
1 mark for 3 correct items
1 mark for correct order with null pointer in last node **[3]**

(b) (i)

Stack

TopOfStackPointer

0

FreePointer

1

	Name	Pointer
[1]		2
[2]		3
[3]		4
[4]		5
[5]		6
[6]		7
[7]		8
[8]		9
[9]		10
[10]		0

Mark as follows:
TopOfStackPointer
FreePointer
Pointers[1] to [9]
Pointer[10]

[4]

(ii)
```
PROCEDURE Pop()
    // Report error if Stack is empty
    IF TopOfStackPointer = 0
      THEN
        Error
      ELSE
        OUTPUT Stack[TopOfStackPointer].Name
        // take a copy of the current top of stack pointer
        TempPointer ← TopOfStackPointer
        // update the top of stack pointer
        TopOfStackPointer ← Stack[TempPointer].Pointer
        // link released node to free list
        Stack[TempPointer].Pointer ← FreePointer
        FreePointer ← TempPointer
    ENDIF
ENDPROCEDURE
```

1 mark for each line of code as above (first 4 lines + ENDIF for 1 mark) **[Max 5]**

6 (a) A procedure that calls itself // is defined in terms of itself **[1]**

(b) Before procedure call is executed current state of the registers/local variables is saved onto the stack
When returning from a procedure call the registers/local variables are re-instated **[2]**

(c)

Call number	n	(n=0) OR (n=1)	n DIV 2	n MOD 2
1	40	FALSE	20	0
2	20	FALSE	10	0
3	10	FALSE	5	0
4	5	FALSE	2	1
5	2	FALSE	1	0
6	1	TRUE		
		1 mark	*1 mark*	*1 mark*

OUTPUT 101000 – 1 mark for each pair of bits. **[6]**

(d) Conversion of denary number into binary **[1]**

(e) (i) Example Pascal

```
Procedure X(n: INTEGER)
BEGIN
  IF (n = 0) OR (n = 1)
    THEN
        Write(n)
    ELSE
      BEGIN
        X(n DIV 2);
        Write(n MOD 2);
      END;
END;
```

Example Python

```
def X(n):
   if (n == 0) or (n == 1):
      print(n, end="")
   else:
      X(n // 2)
      print(n % 2, end="")
```

Mark as follows:
Procedure heading & ending
Boolean expression
correctly grouped statements within ELSE
recursive call
Using DIV and MOD correctly **[5]**