# Chapter 24 Algorithm Design Methods: Answers to coursebook questions and tasks

Syllabus sections covered: 4.2

## Task 24.01

```
PROGRAM ProcessOrderForm
PROCEDURE ProcessHeaderData
   INPUT CustomerName
   INPUT CustomerAddress
ENDPROCEDURE

PROCEDURE ProcessOrderData
   REPEAT
      INPUT ProductID
      INPUT Quantity
   UNTIL no more products
ENDPROCEDURE

PROCEDURE CalculateTotals
   Total ← ProductTotal + PostageAndPacking
ENDPROCEDURE

PROCEDURE ReadPaymentMethods
   IF ChequePayment = TRUE
      THEN
         ProcessCheque
      ELSE
         READ BankCardDetails
ENDPROCEDURE

CALL ProcessHeaderData
CALL ProcessOrderData
CALL CalculateTotals
CALL ReadPaymentMethods
ENDPROGRAM
```

## Task 24.02

**Python**

```
# status can be one of:

INACTIVE = "System inactive"
ACTIVE = "System active"
ALERT = "alert mode"
ALARM = "Alarm bell ringing"

def PressStart(state) :
   if state == INACTIVE :
      state = ACTIVE
   print(state)
   return(state)

def EnterPin(state, timer) :
   if state == ACTIVE :
      state = INACTIVE
   elif state == ALERT :
      state = INACTIVE
      timer = 0
   elif state == ALARM :
      state = INACTIVE
   print(state)
   return(state, timer)

def ActivateSensor(state) :
   if state == ACTIVE :
      state = ALERT
   print(state)
   return(state)

def IncTimer(state, timer) :
   if state == ALERT :
      timer += 1
   return(timer)

def StartBell(state, timer) :
   if state == ALERT and timer == 2 :
      state = ALARM
   print(state)
   return(state)

def ListEvents() :
   print("1 : Start button")
   print("2: Enter PIN")
   print("3: Activate Sensor")

def Wait() :
   for i in range(2000) :
      # do nothing
      print(end='')

def main() :
   state = INACTIVE
   timer = 0
   print(state)
   while True :
      ListEvents()
      Event = input("Enter Event: ")
      if Event == "1" :
         state = PressStart(state)
      elif Event == "2" :
         state, timer = EnterPin(state, timer)
      elif Event == "3" :
         state = ActivateSensor(state)
```

```
        Wait()
        timer = IncTimer(state, timer)
        state = StartBell(state, timer)

main()
```

**VB.NET**

```
Module Module1
    ' status can be one of:
    Const INACTIVE = "System inactive"
    Const ACTIVE = "System active"
    Const ALERT = "alert mode"
    Const ALARM = "Alarm bell ringing"

    Sub PressStart(ByRef state As String)
        If state = INACTIVE Then
            state = ACTIVE
        End If
        Console.WriteLine(state)
    End Sub

    Sub EnterPin(ByRef state As String, ByRef timer As Integer)
        If state = ACTIVE Then
            state = INACTIVE
        Else
            If state = ALERT Then
                state = INACTIVE
                timer = 0
            Else
                If state = ALARM Then
                    state = INACTIVE
                End If
            End If
        End If
        Console.WriteLine(state)
    End Sub

    Sub ActivateSensor(ByRef state As String)
        If state = ACTIVE Then
            state = ALERT
        End If
        Console.WriteLine(state)
    End Sub

    Sub IncTimer(ByVal state As String, ByRef timer As Integer)
        If state = ALERT Then
            timer = timer + 1
        End If
    End Sub

    Sub StartBell(ByRef state As String, ByVal timer As Integer)
        If (state = ALERT) And (timer = 2) Then
            state = ALARM
        End If
        Console.WriteLine(state)
    End Sub

    Sub ListEvents()
        Console.WriteLine("1 : Start button")
        Console.WriteLine("2: Enter PIN")
        Console.WriteLine("3: Activate Sensor")
    End Sub
    Sub Wait()
        For i = 1 To 2000
            ' do nothing
            Console.Write("")
        Next
    End Sub

    Sub Main()
        Dim state As String
        Dim timer As Integer
        Dim EventChosen As Char
        state = INACTIVE
        timer = 0
        Console.WriteLine(state)
        Do While True
            ListEvents()
            Console.Write("Enter Event: ")
            EventChosen = Console.ReadLine()
            Select Case EventChosen
                Case "1"
                    PressStart(state)
                Case "2"
                    EnterPin(state, timer)
                Case "3"
                    ActivateSensor(state)
            End Select
            Wait()
            IncTimer(state, timer)
            StartBell(state, timer)
        Loop
    End Sub

End Module
```

**Pascal**

```
// status can be one of:

const
INACTIVE = 'System inactive';
ACTIVE = 'System active' ;
ALERT = 'alert mode';
ALARM = 'Alarm bell ringing';

procedure PressStart(var state : string);
begin
   if state = INACTIVE then
      state := ACTIVE;
   WriteLn(state);
end;

procedure EnterPin(var state : string; var timer :
integer);
begin
   if state = ACTIVE
   then
      state := INACTIVE
   else
      if state = ALERT
      then
```

```
         begin
           state := INACTIVE;
           timer := 0;
         end
      else
         if state = ALARM
         then
           state := INACTIVE;
   WriteLn(state);
end;

procedure ActivateSensor(var state : string);
begin
   if state = ACTIVE
   then
      state := ALERT;
   WriteLn(state);
end;

procedure IncTimer(state : string; var timer :
integer);
begin
   if state = ALERT
   then
      timer := timer + 1
end;

procedure StartBell(var state : string; timer :
integer);
begin
  if (state = ALERT) and (timer = 2)
  then
      state := ALARM;
   WriteLn(state);
end;

procedure ListEvents;
begin
   WriteLn('1 : Start button');
   WriteLn('2: Enter PIN');
   WriteLn('3: Activate Sensor');
end;

procedure Wait;
var i : integer;
begin
   for i := 1 to 2000 do
      // do nothing
      write('');
end;

procedure main;
var state : string;
    timer : integer;
    Event : char;
```

```
begin
   state := INACTIVE;
   timer := 0;
   WriteLn(state);
   while True do
     begin
       ListEvents;
       write('Enter Event: ');
       Readln(Event);
       Case Event of
         '1' : PressStart(state);
         '2' : EnterPin(state, timer);
         '3' : ActivateSensor(state);
       end;
       Wait;
       IncTimer(state, timer);
       StartBell(state, timer);
     end;
end;

begin
   main;
end.
```

### Question 24.01

1101 (in order of output) but this represents the number 1011

### Extension question 24.01

Yes

### Exam style Questions

1 a

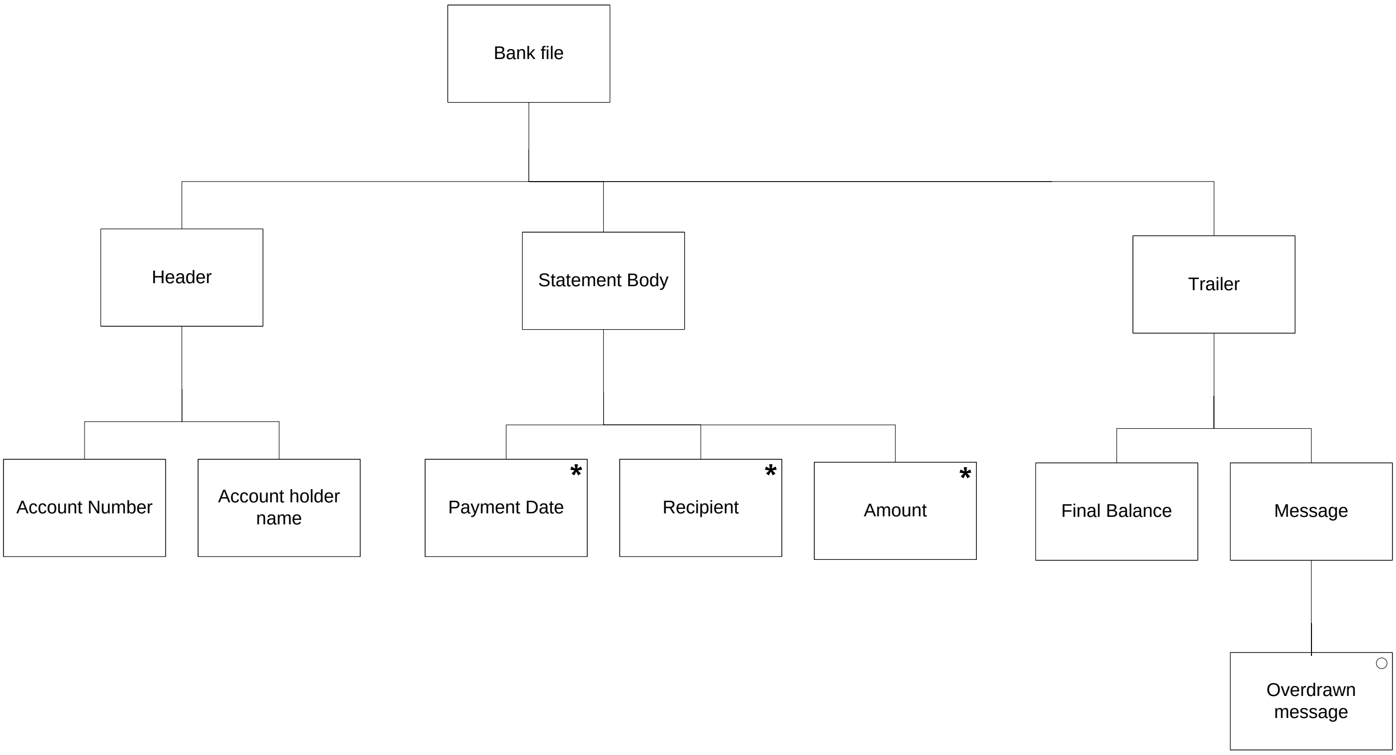
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Conditions** | passenger vehicle | Y | Y | Y | Y | N | N | N | N |
| | between 06:00 and 19:00 | Y | Y | N | N | Y | Y | N | N |
| | more than 3 occupants | Y | N | Y | N | Y | N | Y | N |
| **Actions** | standard charge | | x | | | x | x | x | x |
| | reduced charge | | | | x | | | | |
| | free | x | | x | | | | | |

b

| | | | | | |
|---|---|---|---|---|---|
| Conditions | passenger vehicle | Y | Y | Y | N |
| | between 06:00 and 19:00 | - | Y | N | - |
| | more than 3 occupants | Y | N | N | - |
| Actions | standard charge | | x | | x |
| | reduced charge | | | x | |
| | free | x | | | |

2

Figure 24.01



3

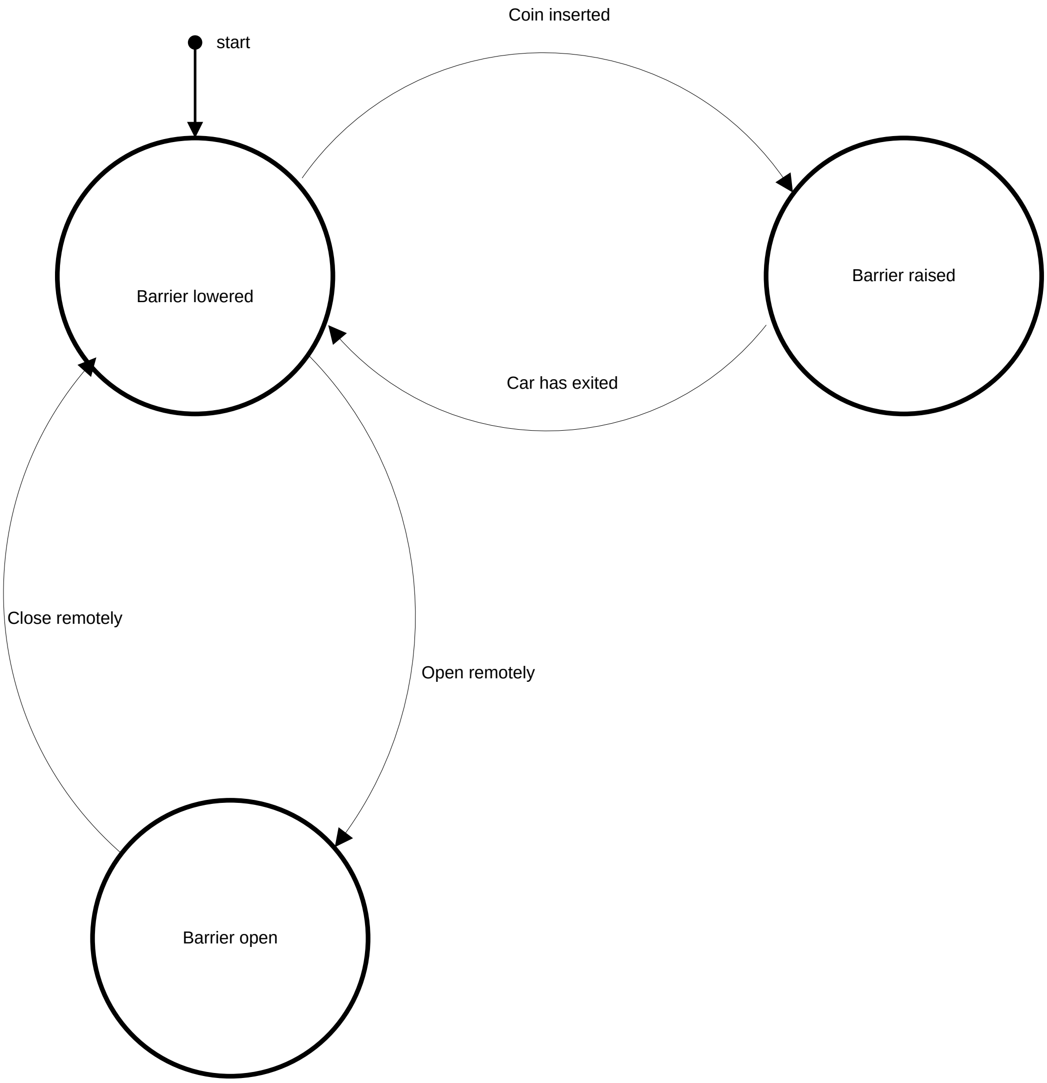


Figure 24.02