



SPECIMEN MATERIAL

AS
COMPUTER SCIENCE
(7516/1E)

Paper 1 VB.net

Skeleton Program

Skeleton Program code for the AQA AS Unit 1 SAM this code should be used in conjunction with the Preliminary Material written by the AQA Programmer Team developed in the Visual Studio 2008 (Console Mode) programming environment VB.NET)

```
Module Module1
    Const NoOfTraps = 2
    Const NSDistance = 5
    Const WEDistance = 7

    Structure CellReference
        Dim NoOfCellsSouth As Integer
        Dim NoOfCellsEast As Integer
    End Structure

    Structure GameData
        Dim TrapPositions() As CellReference
        Dim MonsterPosition As CellReference
        Dim PlayerPosition As CellReference
        Dim FlaskPosition As CellReference
        Dim MonsterAwake As Boolean
    End Structure

    Sub Main()
        Dim Cavern(NSDistance-1, WEDistance-1) As Char
        Dim Choice As Integer
        Dim FlaskPosition As CellReference
        Dim MonsterAwake As Boolean
        Dim MonsterPosition As CellReference
        Dim PlayerPosition As CellReference
        Dim TrapPositions(NoOfTraps) As CellReference
        Randomize()
        Do
            DisplayMenu()
            Choice = GetMainMenuChoice()
            Select Case Choice
                Case 1
                    SetUpGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
                    FlaskPosition, MonsterAwake, True)
                    PlayGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
                    FlaskPosition, MonsterAwake)
                Case 2
                    LoadGame(TrapPositions, MonsterPosition, PlayerPosition, FlaskPosition,
                    MonsterAwake)
                    SetUpGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
                    FlaskPosition, MonsterAwake, False)
                    PlayGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
                    FlaskPosition, MonsterAwake)
                Case 3 : SaveGame(TrapPositions, MonsterPosition, PlayerPosition,
                    FlaskPosition, MonsterAwake)
                Case 4
                    SetUpTrainingGame(Cavern, TrapPositions, MonsterPosition,
                    PlayerPosition, FlaskPosition, MonsterAwake)
            End Select
        Loop
    End Sub
End Module
```

```
        PlayGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
FlaskPosition, MonsterAwake)
    End Select
    Loop Until Choice = 9
End Sub

Sub DisplayMenu()
    Console.WriteLine("MAIN MENU")
    Console.WriteLine()
    Console.WriteLine("1. Start new game")
    Console.WriteLine("2. Load game")
    Console.WriteLine("3. Save game")
    Console.WriteLine("4. Play training game")
    Console.WriteLine("9. Quit")
    Console.WriteLine()
    Console.Write("Please enter your choice: ")
End Sub

Function GetMainMenuChoice() As Integer
    Dim Choice As Integer
    Choice = CInt(Console.ReadLine())
    Console.WriteLine()
    GetMainMenuChoice = Choice
End Function

Sub ResetCavern(ByRef Cavern(,) As Char)
    Dim Count1 As Integer
    Dim Count2 As Integer
    For Count1 = 0 To NSDistance-1
        For Count2 = 0 To WEDistance-1
            Cavern(Count1, Count2) = " "
        Next
    Next
End Sub

Function GetNewRandomPosition() As CellReference
    Dim Position As CellReference
    Do
        Position.NoOfCellsSouth = Int(Rnd() * NSDistance)
        Position.NoOfCellsEast = Int(Rnd() * WEDistance)
    Loop Until Position.NoOfCellsSouth > 0 Or Position.NoOfCellsEast > 0
    'a random coordinate of (0,0) needs to be rejected as this is the starting
position of the player
    GetNewRandomPosition = Position
End Function

Sub SetPositionOfItem(ByRef Cavern(,) As Char, ByRef ObjectPosition As
CellReference, ByVal Item As Char, ByVal NewGame As Boolean)
    Dim Position As CellReference
    If NewGame And Item <> "*" Then
        Do
            Position = GetNewRandomPosition()
        Loop Until Cavern(Position.NoOfCellsSouth, Position.NoOfCellsEast) = " "
        ObjectPosition = Position
    End If
End Sub
```

```
End If
Cavern(ObjectPosition.NoOfCellsSouth, ObjectPosition.NoOfCellsEast) = Item
End Sub
```

```
Sub SetUpGame(ByRef Cavern(,) As Char, ByRef TrapPositions() As CellReference,
ByRef MonsterPosition As CellReference, ByRef PlayerPosition As CellReference,
ByRef FlaskPosition As CellReference, ByRef MonsterAwake As Boolean, ByVal NewGame
As Boolean)
    Dim Count As Integer
    ResetCavern(Cavern)
    If NewGame Then
        PlayerPosition.NoOfCellsSouth = 0
        PlayerPosition.NoOfCellsEast = 0
        MonsterAwake = False
    End If
    For Count = 0 To NoOfTraps-1
        SetPositionOfItem(Cavern, TrapPositions(Count), "T", NewGame)
    Next
    SetPositionOfItem(Cavern, MonsterPosition, "M", NewGame)
    SetPositionOfItem(Cavern, FlaskPosition, "F", NewGame)
    SetPositionOfItem(Cavern, PlayerPosition, "*", NewGame)
End Sub
```

```
Sub SetUpTrainingGame(ByRef Cavern(,) As Char, ByRef TrapPositions() As
CellReference, ByRef MonsterPosition As CellReference, ByRef PlayerPosition As
CellReference, ByRef FlaskPosition As CellReference, ByRef MonsterAwake As
Boolean)
    ResetCavern(Cavern)
    PlayerPosition.NoOfCellsSouth = 2
    PlayerPosition.NoOfCellsEast = 4
    MonsterAwake = False
    TrapPositions(1).NoOfCellsSouth = 1
    TrapPositions(1).NoOfCellsEast = 6
    TrapPositions(2).NoOfCellsSouth = 3
    TrapPositions(2).NoOfCellsEast = 4
    MonsterPosition.NoOfCellsSouth = 0
    MonsterPosition.NoOfCellsEast = 3
    FlaskPosition.NoOfCellsSouth = 4
    FlaskPosition.NoOfCellsEast = 5
    SetUpGame(Cavern, TrapPositions, MonsterPosition, PlayerPosition,
FlaskPosition, MonsterAwake, False)
End Sub
```

```
Sub LoadGame(ByRef TrapPositions() As CellReference, ByRef MonsterPosition As
CellReference, ByRef PlayerPosition As CellReference, ByRef FlaskPosition As
CellReference, ByRef MonsterAwake As Boolean)
    Dim Filename As String
    Dim LoadedGameData As GameData
    Console.WriteLine("Enter the name of the file to load: ")
    Filename = Console.ReadLine
    Console.WriteLine()
    FileOpen(1, Filename, OpenMode.Binary, OpenAccess.Read)
    FileGet(1, LoadedGameData)
```

```
FileClose(1)
TrapPositions = LoadedGameData.TrapPositions
MonsterPosition = LoadedGameData.MonsterPosition
PlayerPosition = LoadedGameData.PlayerPosition
FlaskPosition = LoadedGameData.FlaskPosition
MonsterAwake = LoadedGameData.MonsterAwake
End Sub

Sub SaveGame(ByVal TrapPositions() As CellReference, ByVal MonsterPosition As
CellReference, ByVal PlayerPosition As CellReference, ByVal FlaskPosition As
CellReference, ByVal MonsterAwake As Boolean)
    Dim Filename As String
    Dim CurrentGameData As GameData
    CurrentGameData.TrapPositions = TrapPositions
    CurrentGameData.MonsterPosition = MonsterPosition
    CurrentGameData.PlayerPosition = PlayerPosition
    CurrentGameData.FlaskPosition = FlaskPosition
    CurrentGameData.MonsterAwake = MonsterAwake
    Console.WriteLine("Enter new file name: ")
    Filename = Console.ReadLine
    Console.WriteLine()
    FileOpen(1, Filename, OpenMode.Binary, OpenAccess.Write)
    FilePut(1, CurrentGameData)
    FileClose(1)
End Sub

Sub DisplayCavern(ByVal Cavern(,) As Char, ByVal MonsterAwake As Boolean)
    Dim Count1 As Integer
    Dim Count2 As Integer
    For Count1 = 0 To NSDistance-1
        Console.WriteLine(" ----- ")
        For Count2 = 0 To WEDistance-1
            If Cavern(Count1, Count2) = " " Or Cavern(Count1, Count2) = "*" Or
(Cavern(Count1, Count2) = "M" And MonsterAwake) Then
                Console.Write("|" & Cavern(Count1, Count2))
            Else
                Console.Write("| ")
            End If
        Next
        Console.WriteLine("|")
    Next
    Console.WriteLine(" ----- ")
    Console.WriteLine()
End Sub

Sub DisplayMoveOptions()
    Console.WriteLine()
    Console.WriteLine("Enter N to move NORTH")
    Console.WriteLine("Enter E to move EAST")
    Console.WriteLine("Enter S to move SOUTH")
    Console.WriteLine("Enter W to move WEST")
    Console.WriteLine("Enter M to return to the Main Menu")
    Console.WriteLine()
End Sub
```

```
Function GetMove() As Char
    Dim Move As Char
    Move = Console.ReadLine
    Console.WriteLine()
    GetMove = Move
End Function

Sub MakeMove(ByRef Cavern(,) As Char, ByVal Direction As Char, ByRef
PlayerPosition As CellReference)
    Cavern(PlayerPosition.NoOfCellsSouth, PlayerPosition.NoOfCellsEast) = " "
    Select Case Direction
        Case "N"
            PlayerPosition.NoOfCellsSouth = PlayerPosition.NoOfCellsSouth - 1
        Case "S"
            PlayerPosition.NoOfCellsSouth = PlayerPosition.NoOfCellsSouth + 1
        Case "W"
            PlayerPosition.NoOfCellsEast = PlayerPosition.NoOfCellsEast - 1
        Case "E"
            PlayerPosition.NoOfCellsEast = PlayerPosition.NoOfCellsEast + 1
    End Select
    Cavern(PlayerPosition.NoOfCellsSouth, PlayerPosition.NoOfCellsEast) = "*"
End Sub

Function CheckValidMove(ByVal PlayerPosition As CellReference, ByVal Direction
As Char) As Boolean
    Dim ValidMove As Boolean
    ValidMove = True
    If Not (Direction = "N" Or Direction = "S" Or Direction = "W" Or Direction =
"E" Or Direction = "M") Then
        ValidMove = False
    End If
    CheckValidMove = ValidMove
End Function

Function CheckIfSameCell(ByVal FirstCellPosition As CellReference, ByVal
SecondCellPosition As CellReference) As Boolean
    Dim InSameCell As Boolean
    InSameCell = False
    If FirstCellPosition.NoOfCellsSouth = SecondCellPosition.NoOfCellsSouth And
FirstCellPosition.NoOfCellsEast = SecondCellPosition.NoOfCellsEast Then
        InSameCell = True
    End If
    CheckIfSameCell = InSameCell
End Function

Sub DisplayWonGameMessage()
    Console.WriteLine("Well done! You have found the flask containing the Styxian
potion.")
    Console.WriteLine("You have won the game of MONSTER!")
    Console.WriteLine()
End Sub
```

```
Sub DisplayTrapMessage()
    Console.WriteLine("Oh no! You have set off a trap. Watch out, the monster is
now awake!")
    Console.WriteLine()
End Sub

Sub MoveFlask(ByRef Cavern(,) As Char, ByVal NewCellForFlask As CellReference,
ByRef FlaskPosition As CellReference)
    Cavern(NewCellForFlask.NoOfCellsSouth, NewCellForFlask.NoOfCellsEast) = "F"
    Cavern(FlaskPosition.NoOfCellsSouth, FlaskPosition.NoOfCellsEast) = " "
    FlaskPosition = NewCellForFlask
End Sub

Sub MakeMonsterMove(ByRef Cavern(,) As Char, ByRef MonsterPosition As
CellReference, ByRef FlaskPosition As CellReference, ByVal PlayerPosition As
CellReference)
    Dim OriginalMonsterPosition As CellReference
    Dim MonsterMovedToSameCellAsFlask As Boolean
    OriginalMonsterPosition = MonsterPosition
    Cavern(MonsterPosition.NoOfCellsSouth, MonsterPosition.NoOfCellsEast) = " "
    If MonsterPosition.NoOfCellsSouth < PlayerPosition.NoOfCellsSouth Then
        MonsterPosition.NoOfCellsSouth = MonsterPosition.NoOfCellsSouth + 1
    Else
        If MonsterPosition.NoOfCellsSouth > PlayerPosition.NoOfCellsSouth Then
            MonsterPosition.NoOfCellsSouth = MonsterPosition.NoOfCellsSouth - 1
        Else
            If MonsterPosition.NoOfCellsEast < PlayerPosition.NoOfCellsEast Then
                MonsterPosition.NoOfCellsEast = MonsterPosition.NoOfCellsEast + 1
            Else
                MonsterPosition.NoOfCellsEast = MonsterPosition.NoOfCellsEast - 1
            End If
        End If
    End If
    End If
    MonsterMovedToSameCellAsFlask = CheckIfSameCell(MonsterPosition,
FlaskPosition)
    If MonsterMovedToSameCellAsFlask Then
        MoveFlask(Cavern, OriginalMonsterPosition, FlaskPosition)
    End If
    Cavern(MonsterPosition.NoOfCellsSouth, MonsterPosition.NoOfCellsEast) = "M"
End Sub

Sub DisplayLostGameMessage()
    Console.WriteLine("ARGHHHHHH! The monster has eaten you. GAME OVER.")
    Console.WriteLine("Maybe you will have better luck next time you play
MONSTER!")
    Console.WriteLine()
End Sub

Sub PlayGame(ByRef Cavern(,) As Char, ByVal TrapPositions() As CellReference,
ByRef MonsterPosition As CellReference, ByRef PlayerPosition As CellReference,
ByRef FlaskPosition As CellReference, ByRef MonsterAwake As Boolean)
    Dim Count As Integer
    Dim Eaten As Boolean
    Dim FlaskFound As Boolean
```

```
Dim MoveDirection As Char
Dim ValidMove As Boolean
Eaten = False
FlaskFound = False
DisplayCavern(Cavern, MonsterAwake)
Do
    Do
        DisplayMoveOptions()
        MoveDirection = GetMove()
        ValidMove = CheckValidMove(PlayerPosition, MoveDirection)
    Loop Until ValidMove
    If MoveDirection <> "M" Then
        MakeMove(Cavern, MoveDirection, PlayerPosition)
        DisplayCavern(Cavern, MonsterAwake)
        FlaskFound = CheckIfSameCell(PlayerPosition, FlaskPosition)
        If FlaskFound Then
            DisplayWonGameMessage()
        End If
        Eaten = CheckIfSameCell(MonsterPosition, PlayerPosition)
        If Not MonsterAwake And Not FlaskFound And Not Eaten Then
            MonsterAwake = CheckIfSameCell(PlayerPosition, TrapPositions(1))
            If Not MonsterAwake Then
                MonsterAwake = CheckIfSameCell(PlayerPosition, TrapPositions(2))
            End If
            If MonsterAwake Then
                DisplayTrapMessage()
                DisplayCavern(Cavern, MonsterAwake)
            End If
        End If
        If MonsterAwake And Not Eaten And Not FlaskFound Then
            Count = 0
            Do
                MakeMonsterMove(Cavern, MonsterPosition, FlaskPosition,
PlayerPosition)
                Eaten = CheckIfSameCell(MonsterPosition, PlayerPosition)
                Console.WriteLine()
                Console.WriteLine("Press Enter key to continue")
                Console.ReadLine()
                DisplayCavern(Cavern, MonsterAwake)
                Count = Count + 1
            Loop Until Count = 2 Or Eaten
        End If
        If Eaten Then
            DisplayLostGameMessage()
        End If
    End If
    Loop Until Eaten Or FlaskFound Or MoveDirection = "M"
End Sub
End Module
```
