Cambridge Assessment International Education

Cambridge O Level

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **May/June 2020**

PRE-RELEASE MATERIAL

9124647605

No additional materials are needed.

This material should be given to the relevant teachers and candidates as soon as it has been received at the centre.

INSTRUCTIONS

- You should use this material in preparation for the examination.
- You should attempt the practical programming tasks using your chosen high-level, procedural programming language.

Your preparation for the examination should include attempting the following practical tasks by **writing and testing a program or programs**.

A car park payment system allows customers to select the number of hours to leave their car in the car park. The customer will get a discount if they enter their frequent parking number correctly. The system calculates and displays the amount the customer must pay. The price of parking, the number of hours the customer can enter, and any discount depend upon the day of the week and the arrival time. The number of hours entered is a whole number. The price per hour is calculated using the price in force at the arrival time. No parking is allowed between Midnight and 08:00.

Day of the week	Arrival time			
	From 08:00 to 15:59		From 16:00 to Midnight	
	Max stay in hours	Price per hour	Hours	Price
Sunday	8	2.00	Up to Midnight	2.00
Monday	2	10.00	Up to Midnight	2.00
Tuesday	2	10.00	Up to Midnight	2.00
Wednesday	2	10.00	Up to Midnight	2.00
Thursday	2	10.00	Up to Midnight	2.00
Friday	2	10.00	Up to Midnight	2.00
Saturday	4	3.00	Up to Midnight	2.00

A frequent parking number can be entered for discounted parking. This number consists of 4 digits and a check digit that is calculated using a modulo 11 check digit calculation. A discount of 50% is available for arrival times from 16:00 to Midnight; the discount is 10% at all other arrival times.

Write and test a program or programs to simulate the car park payment system.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Calculating the price to park.

A customer inputs the day, the hour of arrival excluding minutes (for example 15:45 would be 15), the number of hours to leave their car, and a frequent parking number if available. If the frequent parking number has an incorrect check digit, then no discount can be applied. The price to park, based on the day, the hour of arrival, the number of hours of parking required and any discount available, is calculated and displayed.

Task 2 – Keeping a total of the payments.

Extend **Task 1** to keep a daily total of payments made for parking. The daily total is zeroed at the start of the day. For the simulation, each customer inputs the amount paid, this must be greater than or equal to the amount displayed. There is no change given so the amount input may exceed the amount displayed.
Each customer payment is added to the daily total, and this total is displayed at the end of the day.

Task 3 – Making payments fairer.

Customers have complained that sometimes they are being charged too much if they arrive before 16:00 and depart after 16:00. Extend **Task 1** to calculate the price before 16:00, then add the evening charge. For example, a customer arriving at 14:45 on a Sunday and parking for five hours was previously charged 10.00 and would now be charged 6.00

TASK 1

[A customer inputs the day,]

```
Sub Main()
    Dim day As String
    Console.Write("Enter the day: ")
    day = Console.ReadLine()
    Console.WriteLine("Thank you.")
End Sub
```

Validation of day

```
Sub Main()
    Dim day As String
    Dim valid As Boolean
    valid = False
    Do
        Console.Write("Enter the day: ")
        day = Console.ReadLine()
        Select Case day
            Case "Monday", "monday"
                valid = True
            Case "Tuesday", "tuesday"
                valid = True
            Case "Wednesday", "wednesday"
                valid = True
            Case "Thursday", "thursday"
                valid = True
            Case "Friday", "friday"
                valid = True
            Case "Saturday", "saturday"
                valid = True
            Case "Sunday", "sunday"
                valid = True
            Case Else
                Console.WriteLine("Please enter the correct day.")
        End Select
    Loop Until valid
    Console.WriteLine("Thank you.")
    Console.ReadLine()
End Sub
```

Modification to the validation of day:

```
    Dim days() As String = {"sunday", "monday", "tuesday", "wednesday", "thursday", "friday",
                            "saturday"}
    Dim index As Integer
    valid = False
    Do
        Console.Write("Enter the day: ")
        day = LCase(Console.ReadLine())
        index = Array.IndexOf(days, day)
        If Not index = -1 Then
            valid = True
        Else
            Console.WriteLine("Please enter the correct day.")
        End If
    Loop Until valid
    Console.WriteLine("Thank you.")
```

[A customer inputs the day, the hour of arrival excluding minutes (for example 15:45 would be 15),]

```
    Dim valid As Boolean
    Dim hour_arrival As Integer
    valid = False
    Do
        Console.Write("Enter the hour of arrival: ")
        hour_arrival = Console.ReadLine()
        Select Case hour_arrival
            Case 0 To 23
                valid = True
            Case Else
                Console.WriteLine("Please enter the correct hour of arrival.")
```

```
            End Select
        Loop Until valid
        Console.WriteLine("Thank you.")
        Console.ReadLine()
    End Sub
```

[No parking is allowed between Midnight and 08:00.]

```
        valid = False
        Do
            Console.Write("Enter the hour of arrival: ")
            hour_arrival = Console.ReadLine()
            Select Case hour_arrival
                Case 0 To 7
                    Console.WriteLine("Cannot park between 0 and 8.")
                Case 8 To 23
                    valid = True
                Case Else
                    Console.WriteLine("Please enter the correct hour of arrival.")
            End Select
        Loop Until valid
        Console.WriteLine("Thank you.")
```

Improvement using constants:

```
        Const mornstarthour = 8
        Const mornfinishhour = 15
        Const evestarthour = 16
        Const evefinishhour = 23
        Const noparkstarthour = 0
        Const noparkfinishhour = 7
        valid = False
        Do
            Console.Write("Enter the hour of arrival: ")
            hour_arrival = Console.ReadLine()
            Select Case hour_arrival
                Case noparkstarthour To noparkfinishhour
                    Console.WriteLine("Cannot park between 0 and 8.")
                Case mornstarthour To evefinishhour
                    valid = True
                Case Else
                    Console.WriteLine("Please enter the correct hour of arrival.")
            End Select
        Loop Until valid
        Console.WriteLine("Thank you.")
```

[the number of hours to leave their car,]

```
        Console.Write("Enter how many hours to leave your car: ")
        hours_leave = Console.ReadLine()
        Console.WriteLine("Thank you.")
```

Validation of the number of hours to leave a car for parking:

```
        valid = False
        Do
            Console.Write("Enter how many hours to leave your car: ")
            hours_leave = Console.ReadLine()
            n = 24 - hour_arrival
            If hours_leave <= n Then
                valid = False
            Else
                Console.WriteLine("Please enter the hours less than or equal to " & n & ".")
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")
```

Improvement in code:

```
        Dim index, hour_arrival, hours_leave, maxhourspark, n, sum As Integer
        Dim mornmaxhours() As Decimal = {8, 2, 2, 2, 2, 2, 4}
        Dim evemaxhours() As Decimal = {8, 8, 8, 8, 8, 8, 8}
        index = Array.IndexOf(days, day)
        valid = False
```

```
Do
    Console.Write("Enter how many hours to leave your car: ")
    hours_leave = Console.ReadLine()
    maxhourspark = 24 - hour_arrival
    If hour_arrival < evestarthour Then
        If maxhourspark > mornmaxhours(index) Then maxhourspark = mornmaxhours(index)
    Else
        If maxhourspark > evemaxhours(index) Then maxhourspark = evemaxhours(index)
    End If
    If hours_leave <= maxhourspark And hours_leave > 0 Then
        valid = True
    Else
        Console.WriteLine("Please enter the hours less than or equal to " &
                          maxhourspark & ".")
    End If
Loop Until valid
Console.WriteLine("Thank you.")
```

[and a frequent parking number if available.] [if available.]

```
Dim isfpn As Char
valid = False
Do
    Console.Write("Do you have a frequent parking number Y/N: ")
    isfpn = Console.ReadLine()
    Select Case isfpn
        Case "Y", "y", "N", "n"
            valid = True
        Case Else
            Console.WriteLine("Please enter Y or N.")
    End Select
Loop Until valid
Console.WriteLine("Thank you.")
```

Improved code:

```
Dim yesno As Char
valid = False
Do
    Console.Write("Do you have a frequent parking number Y/N: ")
    yesno = LCase(Console.ReadLine())
    Select Case yesno
        Case "y"
            valid = True
            isfpn = True
        Case "n"
            valid = True
            isfpn = False
        Case Else
            Console.WriteLine("Please enter Y or N.")
    End Select
Loop Until valid
Console.WriteLine("Thank you.")
```

[and a frequent parking number if available.]

```
Dim fpn As String
If isfpn Then
    Console.Write("Enter your frequent parking number: ")
    fpn = Console.ReadLine()
End If
```

[This number consists of 4 digits and a check digit]

```
If isfpn Then
    valid = False
    Do
        Console.Write("Enter your frequent parking number: ")
        fpn = Console.ReadLine()
        If Len(fpn) <> 5 Then
            Console.WriteLine("The FPN should be 5 digits.")
        Else
            valid = True
```

```
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")
    End If
```

The following code is an improved version to check for digits and Modulo 11 format:

```
    If isfpn Then
        valid = False
        Do
            Console.Write("Enter your frequent parking number: ")
            fpn = UCase(Console.ReadLine())
            If Not Len(fpn) = 5 Then
                Console.WriteLine("The FPN should be 5 digits.")
            Else
                valid = True
                If Not IsNumeric(fpn.Chars(0)) Then
                    valid = False
                ElseIf Not IsNumeric(fpn.Chars(1)) Then
                    valid = False
                ElseIf Not IsNumeric(fpn.Chars(2)) Then
                    valid = False
                ElseIf Not IsNumeric(fpn.Chars(3)) Then
                    valid = False
                ElseIf Not IsNumeric(fpn.Chars(4)) And Not fpn.Chars(4) = "X" Then
                    valid = False
                End If
                If Not valid Then
                    Console.WriteLine("The FPN must be numeric Modulo 11 format.")
                End If
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")
    End If
```

[that is calculated using a modulo 11 check digit calculation.]
The procedure for calculating the check digit, which may be carried out automatically in a computer, is as follows:
First all digits are multiplied individually with a multiplier. The multiplier corresponds to the position of the digit + 1 from the right. All resulting products are added. The result is then divided by 11. The resulting remainder is subtracted from 11 and results in the check digit.
If result is 11 then the check digit is 0. If result is 10 then the check digit is represented by an "X".
A calculation example:

Digits:	4 4 4 0
Weight:	5 4 3 2
Sum:	20 + 16 + 12 + 0 = 48
Remainder:	48 mod 11 = 4
Check digit:	11 – 4 = 7
FPN:	4 4 4 0 7

```
    Dim isfpn, fpncheck As Boolean
    Dim n, sum As Integer
    If isfpn Then
        sum = 0
        n = Val(fpn.Chars(0))
        n = n * 5
        sum = sum + n
        n = Val(fpn.Chars(1))
        n = n * 4
        sum = sum + n
        n = Val(fpn.Chars(2))
        n = n * 3
        sum = sum + n
        n = Val(fpn.Chars(3))
        n = n * 2
        sum = sum + n
        If fpn.Chars(4) = "X" Then
            n = 10
```

```
            Else
                n = Val(fpn.Chars(4))
            End If
            n = n * 1
            sum = sum + n
            If sum Mod 11 = 0 Then
                fpncheck = True
            Else
                fpncheck = False
            End If
        End If
        If isfpn Then
            If fpncheck Then
                Console.WriteLine("Your FPN is accepted and eligible for discount.")
            Else
                Console.WriteLine("Your FPN is rejected and ineligible for discount.")
            End If
        Else
            Console.WriteLine("Without an FPN there is no discount.")
        End If
```

[The price to park, based on the day, the hour of arrival, the number of hours of parking required and any discount available, is calculated and displayed.] [each customer inputs the amount paid, this must be greater than or equal to the amount displayed.]

```
            valid = False
            Do
                Console.Write("Enter the amount paid: ")
                amount_paid = Console.ReadLine()
                If amount_paid >= price Then
                    valid = True
                    daily_total = daily_total + amount_paid
                Else
                    Console.WriteLine("Please enter an amount greater or equal to " & price &
                                      ".")
                End If
            Loop Until valid
            Console.WriteLine("Thank you.")
```

[Task 3: calculate the price before 16:00, then add the evening charge.]

```
            mornhourspark = evestarthour - hour_arrival
            evehourspark = hours_leave - mornhourspark
            If evehourspark < 0 Then
                mornhourspark = mornhourspark + evehourspark
                evehourspark = 0
            End If
```

Task 3:

```
            mornprice = mornhourspark * mornprices(index)
            eveprice = evehourspark * eveprices(index)
            If isfpn And fpncheck Then
                mornprice = mornprice * morndiscount
                eveprice = eveprice * evediscount
            End If
            price = mornprice + eveprice

            If isfpn Then
                If fpncheck Then
                    Console.WriteLine("Your FPN was accepted and discounted price is " & price
& ".")
                Else
                    Console.WriteLine("Your FPN was rejected and undiscounted price is " &
price & ".")
                End If
            Else
                Console.WriteLine("Your undiscounted price without FPN is " & price & ".")
            End If
```

Complete code:

```
Module Module1
```

```
    Sub Main()
        Dim days() As String = {"sunday", "monday", "tuesday", "wednesday", "thursday",
"friday", "saturday"}
        Dim mornmaxhours() As Decimal = {8, 2, 2, 2, 2, 2, 4}
        Dim evemaxhours() As Decimal = {8, 8, 8, 8, 8, 8, 8}
        Dim mornprices() As Decimal = {2, 10, 10, 10, 10, 10, 3}
        Dim eveprices() As Decimal = {2, 2, 2, 2, 2, 2, 2}
        Const mornstarthour = 8
        Const mornfinishhour = 15
        Const evestarthour = 16
        Const evefinishhour = 23
        Const noparkstarthour = 0
        Const noparkfinishhour = 7
        Const morndiscount = 0.9
        Const evediscount = 0.5
        Dim day As String
        Dim valid, isfpn, fpncheck, done As Boolean
        Dim index, hour_arrival, hours_leave, maxhourspark, n, sum As Decimal
        Dim mornhourspark, evehourspark, mornprice, eveprice As Decimal
        Dim price, daily_total, amount_paid As Decimal
        Dim yesno As Char
        Dim input, fpn As String

        valid = False
        Do
            Console.Write("Enter the day: ")
            day = LCase(Console.ReadLine())
            index = Array.IndexOf(days, day)
            If Not index = -1 Then
                valid = True
            Else
                Console.WriteLine("Please enter the correct day.")
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")

        valid = False
        Do
            Console.Write("Any entries Y/N: ")
            yesno = LCase(Console.ReadLine())
            Select Case yesno
                Case "n"
                    valid = True
                    done = True
                Case "y"
                    valid = True
                    done = False
                Case Else
                    Console.WriteLine("Please enter Y or N.")
            End Select
        Loop Until valid

        daily_total = 0

        While Not done

            valid = False
            Do
                Console.Write("Enter the hour of arrival: ")
                input = Console.ReadLine()
                If Not IsNumeric(input) Then
                    Console.WriteLine("Please enter a number.")
                Else
                    hour_arrival = Val(input)
                    Select Case hour_arrival
                        Case noparkstarthour To noparkfinishhour
                            Console.WriteLine("Cannot park between 0 and 8.")
                        Case mornstarthour To evefinishhour
                            valid = True
                        Case Else
```

```
                        Console.WriteLine("Please enter the correct hour of arrival.")
                End Select
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")

        index = Array.IndexOf(days, day)
        valid = False
        Do
            Console.Write("Enter how many hours to leave your car: ")
            input = Console.ReadLine()
            If Not IsNumeric(input) Then
                Console.WriteLine("Please enter a number.")
            Else
                hours_leave = Val(input)
                maxhourspark = 24 - hour_arrival
                If hour_arrival < evestarthour Then
                    If maxhourspark > mornmaxhours(index) Then maxhourspark =
mornmaxhours(index)
                Else
                    If maxhourspark > evemaxhours(index) Then maxhourspark =
evemaxhours(index)
                End If
                If hours_leave <= maxhourspark And hours_leave > 0 Then
                    valid = True
                Else
                    Console.WriteLine("Please enter the hours less than or equal to " &
maxhourspark & ".")
                End If
            End If
        Loop Until valid
        Console.WriteLine("Thank you.")

        valid = False
        Do
            Console.Write("Do you have a frequent parking number Y/N: ")
            yesno = LCase(Console.ReadLine())
            Select Case yesno
                Case "y"
                    valid = True
                    isfpn = True
                Case "n"
                    valid = True
                    isfpn = False
                Case Else
                    Console.WriteLine("Please enter Y or N.")
            End Select
        Loop Until valid
        Console.WriteLine("Thank you.")

        If isfpn Then
            valid = False
            Do
                Console.Write("Enter your frequent parking number: ")
                fpn = UCase(Console.ReadLine())
                If Not Len(fpn) = 5 Then
                    Console.WriteLine("The FPN should be 5 digits.")
                Else
                    valid = True
                    If Not IsNumeric(fpn.Chars(0)) Then
                        valid = False
                    ElseIf Not IsNumeric(fpn.Chars(1)) Then
                        valid = False
                    ElseIf Not IsNumeric(fpn.Chars(2)) Then
                        valid = False
                    ElseIf Not IsNumeric(fpn.Chars(3)) Then
                        valid = False
                    ElseIf Not IsNumeric(fpn.Chars(4)) And Not fpn.Chars(4) = "X" Then
                        valid = False
                    End If
```

```
                    If Not valid Then
                        Console.WriteLine("The FPN must be numeric Modulo 11 format.")
                    End If
                End If
            Loop Until valid
            Console.WriteLine("Thank you.")
        End If

        If isfpn Then
            sum = 0
            n = Val(fpn.Chars(0))
            n = n * 5
            sum = sum + n
            n = Val(fpn.Chars(1))
            n = n * 4
            sum = sum + n
            n = Val(fpn.Chars(2))
            n = n * 3
            sum = sum + n
            n = Val(fpn.Chars(3))
            n = n * 2
            sum = sum + n
            If fpn.Chars(4) = "X" Then
                n = 10
            Else
                n = Val(fpn.Chars(4))
            End If
            n = n * 1
            sum = sum + n
            If (sum Mod 11) = 0 Then
                fpncheck = True
            Else
                fpncheck = False
            End If
        End If

        REM If hour_arrival < evestarthour Then
        REM    price = hours_leave * mornprice(index)
        REM    If fpncheck Then price = price * morndiscount
        REM Else
        REM    price = hours_leave * eveprice(index)
        REM    If fpncheck Then price = price * evediscount
        REM End If

        mornhourspark = evestarthour - hour_arrival
        evehourspark = hours_leave - mornhourspark
        If evehourspark < 0 Then
            mornhourspark = mornhourspark + evehourspark
            evehourspark = 0
        End If

        mornprice = mornhourspark * mornprices(index)
        eveprice = evehourspark * eveprices(index)
        If isfpn And fpncheck Then
            mornprice = mornprice * morndiscount
            eveprice = eveprice * evediscount
        End If
        price = mornprice + eveprice

        If isfpn Then
            If fpncheck Then
                Console.WriteLine("Your FPN was accepted and discounted price is " & price
& ".")
            Else
                Console.WriteLine("Your FPN was rejected and undiscounted price is " &
price & ".")
            End If
        Else
            Console.WriteLine("Your undiscounted price without FPN is " & price & ".")
        End If
```

```
            valid = False
            Do
                Console.Write("Enter the amount paid: ")
                amount_paid = Console.ReadLine()
                If amount_paid >= price Then
                    valid = True
                    daily_total = daily_total + amount_paid
                Else
                    Console.WriteLine("Please enter an amount greater or equal to " & price &
"." )
                End If
            Loop Until valid
            Console.WriteLine("Thank you.")

            valid = False
            Do
                Console.Write("Any further entries Y/N: ")
                yesno = LCase(Console.ReadLine())
                Select Case yesno
                    Case "n"
                        valid = True
                        done = True
                    Case "y"
                        valid = True
                        done = False
                    Case Else
                        Console.WriteLine("Please enter Y or N.")
                End Select
            Loop Until valid

        End While

        Console.WriteLine("Daily total for " & day & " is " & daily_total & ".")

        Console.ReadLine()
    End Sub

End Module
```