Programming - Sample Assessment Task Solutions

Application 1: Salary calculator

PSEUDOCODE

```
Begin
   Sub Main
        Name ← Call InputName
        Days ← Call InputDays
        Hours ← Call InputHours
        Call Calculate
        Call Display
   End Main

   Function InputName
        Input Name
        Return Name
   End InputName

   Function InputDays
        Input Days
        While Days < 1 or Days > 7
             Print "Number of days invalid. Please input 1 to 7 only."
             Input Days
        End While
        Return Days
   End InputDays

   Function InputHours
        For i ← 1 to days
             Input Hours
             While hours <= 0 or hours >24
                 Print "Invalid hours. Input hours again."
                 Input hours
             End while
        Next
        Return Hours
   End InputHours

   Sub Calculate
        TotalHours ← Sum(Hours)
        Highest ← Max(Hours)
        Salary ← TotalHours * 10
        If TotalHours >= 40 then
             Bonus ← Salary * 0.10
             PostBonus ← Salary + Bonus
        End if
   End Calculate

   Sub Display
        IF Bonus > 0 then
             Print "Hello, " + Name + ", you have worked for " + Days + " days. Total hours worked is "
             + TotalHours + " hours. Your highest number of working hours per day is " + Highest + ".
             Your salary is $" + Salary + ". You qualify for a bonus of $" + Bonus + " and your post-
             bonus salary is $" + PostBonus + "."
        Else
             Print "Hello, " + Name + ", you have worked for " + Days + " days. Total hours worked is "
             + TotalHours + " hours. Your highest number of working hours per day is " + Highest + ".
             Your salary is $" + Salary + ". You do not qualify for a bonus."
   End Display
End
```

VB CODE

```
Module Module1
    'Declare variables used in the application
    Dim Name As String
    Dim Days As Integer
    Dim Hours(7) As Integer
    Dim TotalHours As Integer
    Dim Highest As Integer
    Dim Salary As Decimal
    Dim Bonus As Decimal
    Dim PostBonus As Decimal

    'Main procedure used to call functions and procedures
    Sub Main()
        Name = InputName()
        Days = InputDays()
        Hours = InputHours()
        Calculate()
        Display()
    End Sub

    'Function to input user's name. Name is returned to main procedure
    Function InputName()
        Console.WriteLine("Please input name:")
        Name = Console.ReadLine
        Return Name
    End Function

    'Function to input number of days. Days is validated to only accept 1-7. Days is
     then returned to main procedure
    Function InputDays()
        Console.WriteLine("Please input number of days worked in a week:")
        Days = Console.ReadLine
        While Days < 1 Or Days > 7
            Console.WriteLine("Number of days invalid. Please input 1 to 7 only: ")
            Days = Console.ReadLine
        End While
        Return Days
    End Function

    'Function to input number of hours. Hours is validated and displays an error
     message if hours is < 0 or > 24. Hours is then returned to main procedure
    Function InputHours()
        For i As Integer = 1 To Days
            Console.Write("Please input number of hours worked in day " & i & ": ")
            Hours(i) = Console.ReadLine
            While Hours(i) <= 0 Or Hours(i) > 24
                Console.Write("Invalid hours. Please input number of hours worked in
                day " & i & ": ")
                Hours(i) = Console.ReadLine
            End While
        Next
        Return Hours
    End Function
    'Procedure to calculate total hours, highest, salary. Total hours is evaluated to
     determine bonus and post bonus salary
    Sub Calculate()
        TotalHours = Hours.Sum()
        Highest = Hours.Max()
        Salary = TotalHours * 10
        If TotalHours >= 40 Then
            Bonus = Salary * 0.1
            PostBonus = Salary + Bonus
        End If
    End Sub

    'Procedure to display statement based on bonus
    Sub Display()
        If Bonus > 0 Then
            Console.WriteLine("Hello, " & Name & ", you have worked for " & Days &
            " days. Total hours worked is " & TotalHours & " hours. Your highest
            number of working hours per day is " & Highest & ". Your salary is $" &
            Format(Salary, "0.00") & ". You qualify for a bonus of $" &
            Format(Bonus, "0.00") & " and your post-bonus salary is $" &
            Format(PostBonus, "0.00") & ".")
        Else
            Console.WriteLine("Hello, " & Name & ", you have worked for " & Days &
            " days. Total hours worked is " & TotalHours & " hours. Your highest
            number of working hours per day is " & Highest & ". Your salary is $" &
            Format(Salary, "0.00") & ". You do not qualify for a bonus.")
        End If
    End Sub

End Module
```

TEST TABLE

Test	Test Data	Expected Results	Actual results
InputName function is working correctly	“Jane”	“Jane”	“Jane”
InputDays function is working correctly	Days = 0	"Number of days invalid. Please input 1 to 7 only. Please input number of days worked in a week:"	"Number of days invalid. Please input 1 to 7 only. Please input number of days worked in a week:"
InputDays function is working correctly	Days = 1	Days = 1	Days = 1
InputDays function is working correctly	Days = 7	Days = 7	Days = 7
InputDays function is working correctly	Days = 8	"Number of days invalid. Please input 1 to 7 only. Please input number of days worked in a week:"	"Number of days invalid. Please input 1 to 7 only. Please input number of days worked in a week:"
InputHours function is working correctly	Days = 1 Hours(1) = 0	"Invalid hours. Please input number of hours worked in day 1"	"Invalid hours. Please input number of hours worked in day 1"
InputHours function is working correctly	Days = 1 Hours(1) = 25	"Invalid hours. Please input number of hours worked in day 1"	"Invalid hours. Please input number of hours worked in day 1"
InputHours function is working correctly	Days = 3 Hours(1) = 8 Hours(2) = 10 Hours(3) = 6	Hours(1) = 8 Hours(2) = 10 Hours(3) = 6	Hours(1) = 8 Hours(2) = 10 Hours(3) = 6

Calculate procedure is working correctly	Days = 3
	Hours(1) = 16
	Hours(2) = 14
	Hours(3) = 9
		Total hours = 39
		Highest = 16
		Salary = 390
			Total hours = 39
			Highest = 16
			Salary = 390

Calculate procedure is working correctly	Days = 4
	Hours(1) = 12
	Hours(2) = 10
	Hours(3) = 10
	Hours(4) = 9
		Total hours = 40
		Highest = 12
		Salary = 400
		Bonus = 40
		PostBonus = 440
			Total hours = 40
			Highest = 12
			Salary = 400
			Bonus = 40
			PostBonus = 440

Salary Calculator application is working correctly	Name = “Jane” Days = 3 Hours(1) = 16 Hours(2) = 14 Hours(3) = 9	“Hello, Jane, you have worked for 3 days. Total hours worked is 39 hours. Your highest number of working hours per day is 10. Your salary is $390.00. You do not qualify for a bonus.”	“Hello, Jane, you have worked for 3 days. Total hours worked is 39 hours. Your highest number of working hours per day is 10. Your salary is $390.00. You do not qualify for a bonus.”
Salary Calculator application is working correctly	Name = “Jane” Days = 4 Hours(1) = 12 Hours(2) = 10 Hours(3) = 10 Hours(4) = 9	Hello, Jane, you have worked for 4 days. Total hours worked is 40 hours. Your highest number of working hours per day is 12. Your salary is $400.00. You qualify for bonus of $41.00 and your post-bonus salary is $440.00.”	Hello, Jane, you have worked for 4 days. Total hours worked is 40 hours. Your highest number of working hours per day is 12. Your salary is $400.00. You qualify for bonus of $41.00 and your post-bonus salary is $440.00.”

Application 2: Customer Sales

PSEUDOCODE

```
Begin
   Sub Main
      CusID ← Call InputCusID
      Average ← Call InputSales
      Reward ← Call Evaluate
      Call Display
   End Main

   Function InputCusID
      Input CusID
      While CusID length <> 6
         Print “Customer ID must be 6 characters. Please input again.”
         Input CusID
      End While
      Convert CusID to uppercase
      Return CusID
   End InputCusID

   Sub InputSales
      Input Sales(1)
      Input Sales(2)
      Input Sales(3)
      Average ← (Sales(1) + Sales(2) + Sales(3)) / 3
      Return Average
   End Input

   Function Evaluate
      If Average > 1000 then
         Reward ← True
      Else
         Reward ← False
      End If
      Return Reward
   End Evaluate

   Sub Display
      IF Reward = True then
         Print “Customer with ID “ + CusID + “, has sales amount $” + Sales1 + “, $” + Sales2 + “
         and $“ + Sales3 + “. The average sales amount is $“ + Average + “. You will receive a
         reward.”
      Else
         Print “Customer with ID “ + CusID + “, has sales amount $” + Sales1 + “, $” + Sales2 + “
         and $“ + Sales3 + “. The average sales amount is “ + Average + “. You will not receive a
         reward.”
      End If
   End Display
End
```

VB CODE

```
Module Module1
    'Declare variables used in the application
    Dim CusID As String
    Dim sales(3) As Decimal
    Dim average As Decimal
    Dim reward As Boolean

    Sub Main()
        'Input customer ID
        CusID = inputCusID()

        'Input sales & calculate average
        average = inputSales()

        'Evaluate reward
        reward = evaluate()

        'Display statement
        display()
    End Sub

    'Function to input CusID and test if it is a valid length
    Function inputCusID()
        'User inputs customer ID
        Console.Write("Please input your customer ID: ")
        CusID = UCase(Console.ReadLine)

        'Length of CusID is validated using length check
        While Len(CusID) <> 6
            Console.WriteLine("The customer ID length is not valid. Please input
            customer ID again: ")
            CusID = UCase(Console.ReadLine)
        End While

        'Return CusID to main procedure
        Return CusID
    End Function

    'Function to input sales amount and calculate average
    Function inputSales()
        'User inputs 3 sales amounts
        Console.Write("Input sales amount 1: ")
        sales(1) = Console.ReadLine()
        Console.Write("Input sales amount 2: ")
        sales(2) = Console.ReadLine()
        Console.Write("Input sales amount 3: ")
        sales(3) = Console.ReadLine()

        'Calculate average
        average = (sales(1) + sales(2) + sales(3)) / 3

        'Return average to main procedure
        Return average
    End Function
```

```
     Function evaluate()
        'Test whether customer will get a reward
        If average > 1000 Then
            reward = True
        Else
            reward = False
        End If

        'Return reward to main procedure
        Return reward
    End Function

    'Procedure to display statement based on reward
    Sub display()
        If reward = True Then
            Console.WriteLine("Customer with ID " & CusID & ", has sales amount $" &
            Format(sales(1), "0.00") & ", $" & Format(sales(2), "0.00") & " and $" &
            Format(sales(3), "0.00") & ". The average sales amount is $" &
            Format(average, "0.00") & ". You will receive a reward.")
        Else
            Console.WriteLine("Customer with ID " & CusID & ", has sales amount $" &
            Format(sales(1), "0.00") & ", $" & Format(sales(2), "0.00") & " and $" &
            Format(sales(3), "0.00") & ". The average sales amount is $" &
            Format(average, "0.00") & ". You will not receive a reward.")
        End If
        Console.WriteLine()
    End Sub

End Module
```

TEST TABLE

Test	Test Data	Expected Results	Actual results
InputCusID function is working correctly	cs111	"The customer ID length is not valid. Please input customer ID again:"	"The customer ID length is not valid. Please input customer ID again:"
InputCusID function is working correctly	cst111	CusID = "CST111"	CusID = "CST111"
InputSales function is working correctly	sales(1) = 1500 sales(2) = 500 sales(3) = 1000	Average = 1000	Average = 1000
Evaluate function is working correctly	Average = 1001	Reward = True	Reward = True
Evaluate function is working correctly	Average = 1000	Reward = False	Reward = False
Customer Sales application is working correctly	CusID = "CST111" sales(1) = 1500 sales(2) = 500 sales(3) = 1000	"Customer with ID CST111, has sales amount $1500.00, $500.00 and $1000.00.The average sales amount is $1000. You will not receive a reward."	"Customer with ID CST111, has sales amount $1500.00, $500.00 and $1000.00.The average sales amount is $1000. You will not receive a reward."
Customer Sales application is working correctly	CusID = "CST111" sales(1) = 1500 sales(2) = 500 sales(3) = 1002	"Customer with ID CST111, has sales amount $1500.00, $500.00 and $1002.00.The average sales amount is $1001. You will receive a reward."	"Customer with ID CST111, has sales amount $1500.00, $500.00 and $1002.00.The average sales amount is $1001. You will receive a reward."

Application 3: Unit Results

PSEUDOCODE

```
Begin
    Sub Main
        Call inputDetails
        test = Call inputMarks
        Call evaluate
        Call display()
    End Main

    Sub inputDetails
        Input name
        Convert name to uppercase
        Input ID
        While ID is not a number Or ID length <> 8
            Input ID
        End While
        Input email
        Convert email to lowercase
        While email does not contain "@imail.sunway.edu.my"
            Input email
            Convert email to lowercase
        End While
    End inputDetails

    Function inputMarks
        For i ← 1 To 4
            Input testMark
            While testMark < 0 Or testMark > 100
                Input testMark
            End While
        Next
        Return testMark
    End inputMarks

    Sub evaluate
        totalMark = (test(1) * 0.25) + (test(2) * 0.25) + (test(3) * 0.25) + (test(4) * 0.25)
        If totalMark >= 80 Then
            grade = "HD"
        ElseIf totalMark >= 70 Then
            grade = "D"
        ElseIf totalMark >= 60 Then
            grade = "C"
        ElseIf totalMark >= 50 Then
            grade = "P"
        Else
            grade = "N"
        End If
    End evaluate
    Sub display
        Print "Student name: " & Name
        Print "Student ID: " & ID
        Print"Student email: " & email)
        Print "Test 1: " & test(1) & "; Test 2: " & test(2) & "; Test 3: " & test(3) & "; Test 4: " & test(4)
        Print "Overall unit mark: " & totalMark & "% (Grade: " & grade & ")"
        If grade = "HD" Then
            Print Name & ", you qualify for a scholarship. Congratulations!"
        End if
        If grade = "N" Then
            Print Name & ", you need to repeat this unit next semester. Please work harder."
        End if
    End display
End
```

VB CODE

```
Module Module1
    'Declare variables used in the application
    Dim Name As String
    Dim ID As String
    Dim email As String
    Dim totalMark As Integer
    Dim grade As String
    Dim test(4) As Integer

    'Main procesure to call functions and procedures
    Sub Main()
        'Call inputDetails procedure to input student name, ID and email
        inputDetails()

        'Call inputmarks function to obtain all test marks
        test = inputMarks()

        'Call evaluate procedure to calculate total mark and obtain grade
        evaluate()

        'Call display procedure to output statement
        display()
    End Sub

    'Procedure to input student name and change it to uppercase
    Sub inputDetails()
        'User inputs student name and converted to uppercase
        Console.Write("Input student name: ")
        Name = UCase(Console.ReadLine)

        'User inputs student ID and validated using type check and length check
        Console.Write("Input student ID: ")
        ID = Console.ReadLine
        While IsNumeric(ID) = False Or Len(ID) <> 8
            Console.Write("Student ID must be 8 numbers. Input student ID: ")
            ID = Console.ReadLine
        End While

        'User inputs student email and validated to contain a valid email and converted to lowercase
        Console.Write("Input student email: ")
        email = LCase(Console.ReadLine)
        While email.Contains("@imail.sunway.edu.my") = False
            Console.Write("Email must contain ""@imail.sunway.edu.my"". Input student
              email: ")
            email = LCase(Console.ReadLine)
        End While
    End Sub

    Function inputMarks()
        'Input 4 test marks and validate each mark using a range check
        For i As Integer = 1 To 4
            Console.Write("Test " & i & " mark: ")
            test(i) = Console.ReadLine
            While test(i) < 0 Or test(i) > 100
                Console.Write("Invalid mark. Mark must be 0 - 100.")
                Console.WriteLine("Enter Test " & i & " mark: ")
                test(i) = Console.ReadLine
            End While
        Next

        'Return test marks to main procedure
        Return test
    End Function

    Sub evaluate()
        'Calculate total mark
        totalMark = (test(1) * 0.25) + (test(2) * 0.25) + (test(3) * 0.25) + (test(4)
         * 0.25)

        'evaluate total mark to determine grade
        If totalMark >= 80 Then
            grade = "HD"
        ElseIf totalMark >= 70 Then
            grade = "D"
        ElseIf totalMark >= 60 Then
            grade = "C"
        ElseIf totalMark >= 50 Then
            grade = "P"
        Else
            grade = "N"
        End If
    End Sub

    Sub display()
        'Display summary of student details and marks
        Console.WriteLine()
        Console.WriteLine("Student name: " & Name)
        Console.WriteLine("Student ID: " & ID)
        Console.WriteLine("Student email: " & email)
        Console.WriteLine("Test 1: " & test(1) & "; Test 2: " & test(2) & "; Test 3: "
         & test(3) & "; Test 4: " & test(4))
        Console.WriteLine("Overall unit mark: " & totalMark & "% (Grade: " & grade
         & ")")

        'Display statement based on grade
        If grade = "HD" Then
            Console.WriteLine(Name & ", you qualify for a scholarship.
            Congratulations!")
            Console.WriteLine()
        End If
        If grade = "N" Then
            Console.WriteLine(Name & ", you need to repeat this unit next semester.
            Please work harder.")
            Console.WriteLine()
        End If
    End Sub
End Module
```

Test table

Test	Test Data	Expected Results	Actual results
InputDetails procedure is working correctly (to input name)	Name = “john doe”	Name = “JOHN DOE”	Name = “JOHN DOE”
InputDetails procedure is working correctly (to input student ID)	ID = “1234567”	"Student ID must be 8 numbers. Input student ID: "	"Student ID must be 8 numbers. Input student ID: "
InputDetails procedure is working correctly (to input student ID)	ID = “aaaaaaa”	"Student ID must be 8 numbers. Input student ID: "	"Student ID must be 8 numbers. Input student ID: "
InputDetails procedure is working correctly (to input student email)	email = “JDoe@gmail.com”	“Email must contain "@imail.sunway.edu.my". Input student email: "	“Email must contain "@imail.sunway.edu.my". Input student email: "
InputDetails procedure is working correctly (to input student email)	email = “JDoe@imail.sunway.edu.my”	email = “jdoe@imail.sunway.edu.my”	email = “jdoe@imail.sunway.edu.my”
inputMarks function is working correctly	Test(1) = -1	"Invalid mark. Mark must be 0 - 100." "Enter Test 1 mark: "	"Invalid mark. Mark must be 0 - 100." "Enter Test 1 mark: "
inputMarks function is working correctly	Test(1) = 0 Test(2) = 50 Test(3) = 46 Test(4) = 45	Test(1) = 0 Test(2) = 50 Test(3) = 46 Test(4) = 45	Test(1) = 0 Test(2) = 50 Test(3) = 46 Test(4) = 45
inputMarks function is working correctly	Test(1) = 100 Test(2) = 98 Test(3) = 99 Test(4) = 100	Test(1) = 100 Test(2) = 98 Test(3) = 99 Test(4) = 100	Test(1) = 100 Test(2) = 98 Test(3) = 99 Test(4) = 100
inputMarks function is working correctly	Test(1) = 101	"Invalid mark. Mark must be 0 - 100." "Enter Test 1 mark: "	"Invalid mark. Mark must be 0 - 100." "Enter Test 1 mark: "
Unit Results application is working correctly	Name = “john doe” ID = “18756483” e-mail = “JDOE@imail.sunway.edu.my” Test(1) = 97 Test(2) = 85 Test(3) = 62 Test(4) =76	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 97; Test 2: 85; Test 3: 62; Test 4: 76 Overall unit mark: 80% (Grade: HD) JOHN DOE, you qualify for a scholarship. Congratulations!	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 97; Test 2: 85; Test 3: 62; Test 4: 76 Overall unit mark: 80% (Grade: HD) JOHN DOE, you qualify for a scholarship. Congratulations!

Test	Test Data	Expected Results	Actual results
Unit Results application is working correctly	Name = “john doe” ID = “18756483” e-mail = “JDOE@imail.sunway.edu.my” Test(1) = 94 Test(2) = 85 Test(3) = 62 Test(4) =76	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 94; Test 2: 85; Test 3: 62; Test 4: 76 Overall unit mark: 79% (Grade: D)	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 94; Test 2: 85; Test 3: 62; Test 4: 76 Overall unit mark: 79% (Grade: D)
Unit Results application is working correctly	Name = “john doe” ID = “18756483” e-mail = “JDOE@imail.sunway.edu.my” Test(1) = 48 Test(2) = 50 Test(3) = 50 Test(4) =49	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 48; Test 2: 50; Test 3: 50; Test 4: 49 Overall unit mark: 49% (Grade: N) JOHN DOE, you need to repeat this unit next semester. Please work harder.	Student name: JOHN DOE Student ID: 18756483 Student e-mail: jdoe@imail.sunway.edu.my Test 1: 48; Test 2: 50; Test 3: 50; Test 4: 49 Overall unit mark: 49% (Grade: N) JOHN DOE, you need to repeat this unit next semester. Please work harder.