SLIATE

SRI LANKA INSTITUTE OF ADVANCED TECHNOLOGICAL EDUCATION

(Established in the Ministry of Higher Education, vide in Act No. 29 of 1995)

Higher National Diploma in Information Technology

Second Year, First Semester Examination – 2019

HNDIT2311- Rapid Application Development

Marking Scheme

1)

a) Differentiate Rapid Application Development model from Waterfall model. [Marks 04]

Waterfall	RAD
User involvement is low.	User involvement is high.
It is suitable for well-defined user requirement at the beginning.	It is suitable for small project with changing user requirement.
It is not people centred and incremental approach.	It is people centred and incremental approach.
There is high amount risk in waterfall model.	There is low amount risk in RAD model.
In waterfall model large team size is required	In RAD model small hybrid team size is required.
Any changes can be made in waterfall model only at the beginning.	Any changes can be made in RAD model anytime.
The product of Waterfall model is delivered after the completion of all stages.	The product of RAD model is delivered as soon as possible
There is long waiting time for running software in waterfall model.	There is less waiting time for running software in RAD model, as its first version is released as soon as possible.

b) Timeboxing is one of the feature in RAD model. Illustrate the statement. [Marks 03]
An approach of fixing the resource allocation for a project or a part of a project
Secondary features may be dropped in order to stay on schedule

c) State four-parts of strategy in Rapid Application Development. [Marks 04]

a. Avoid classic mistakes
b. Apply development fundamentals
c. Manage risks to avoid catastrophic setbacks
d. Apply schedule-oriented practices

d) What is meant by classic mistake and how do you prevent people related mistakes.
[Marks 04]

To achieve rapid development, you need to avoid making any big mistakes. Some inefficient development practices have been chosen so often with such predictable bad results that they deserve to be called "classic mistakes". [Marks 01]

- Motivate project team without undermining [Marks 03]
- Choose multi skilled hybrid team
- Solving employee issues / problems quickly and instantly
- Avoid people with heroics character
- Provide peaceful environment with necessary facilities
- Avoid to adding people to a late project
- Maintain good communication between customer and developers

e) Define Integrated Development Environment and three characteristics of it. [Marks 05]

Integrated Development Environment – allows the automation of many of the common programming tasks in one environment. [Marks 01]

i. Writing code [Marks 04]
ii. Checking for syntax (language) errors
iii. Compiling and interpreting (transferring to computer language)
iv. Debugging (fixing run-time or logic errors)
v. Running application

2)

a) Explain scope of variable using suitable example. [Marks 04]

i. Local variable [Marks 02]
Variable declared within a code block is only visible to statements within that code block.

ii. Global variable [Marks 02]
Can be declared at the beginning of the class code window (general declarations section) and be available to all blocks.

b) Give four (4) logical operators in their precedence order (highest to lowest).

– Not
– And
– Or
– Xor

[Marks 04]

c) Write syntax for *if else* control statement. [Marks 03]

```
If      condition   Then
statement[s]
elseIf  condition   Then
statement[s]
else
statement[s]
End If
```

d) Compare *do while* loop and *do until* loop. [Marks 03]

Do while: Do the body of the loop while the condition is true. If the condition is true, the body of the loop is executed.

Do until: Do the body of the loop until the condition is true. If the condition is false, the body of the loop is executed.

e) Write output of the following code segments. [Marks 06]

i.
```
For i = 6 To 13 Step 3
Console.WriteLine(i & "    " & i * i)
Next
```

```
6  36
9  81
12  144
```

ii.
```
Dim n As Integer = 10
Do
Console.WriteLine (n + 5)
n = n - 2
Loop While (n > 1)
```

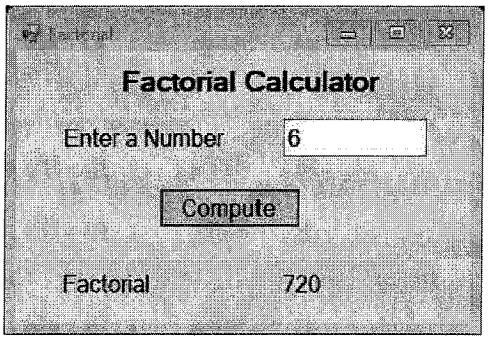
```
15
13
11
9
7
```

iii.
```
Dim a As Short = 5
Do
Console.WriteLine (a)
a += a
Loop Until (a = 15)
```

```
5
10
20
```

3)

a) Consider the bfollowing interface and provide VB.NET code segment for calculating Factoria of an entered number as you press "Compute" button using function with passing parameter. [Marks 08]




```
Private Sub Btnclick_Click(sender As Object, e As EventArgs) Handles
btnclick.Click
        Dim f As Integer = calFact(txtnum.Text)                    [Marks 02]
        lblfact.Text = f                                           [Marks 0.5]
    End Sub

    Function calFact(ByVal a As Integer)                           [Marks 02]
        Dim fact As Integer = 1                                    [Marks 0.5]
        Do
            fact *= a                                              [Marks 01]
            a = a - 1                                              [Marks 01]
        Loop While (a > 1)                                         [Marks 01]
        Return fact
    End Function
```

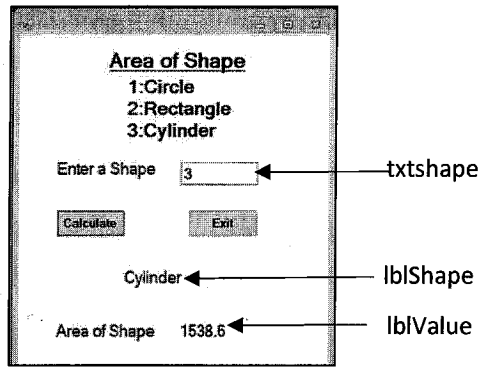
b) An application is created using VB.NET for identifying passed students who got marks greate than 50. Write a code for it. [Marks 08]

i. Store following student marks in a array,
   Marks → 56, 78, 45, 67, 80, 67
ii. Count number passed of students
iii. Calculate total marks of passed students

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
Dim marks() As Integer = {56, 78, 45, 67, 80, 67}                  [Marks 02]
        Dim total, count As Integer                                [Marks 0.5]
        For i = 0 To 5 Step 1                                      [Marks 0.5]
            If (marks(i) >= 50) Then                               [Marks 01]
                total += marks(i)                                  [Marks 02]
                count += 1                                         [Marks 01]
            End If
        Next
        MsgBox("Number of Passed Student" & count)                 [Marks 0.5]
        MsgBox("Total Marks of Passed Student" & total)            [Marks 0.5]
    End Sub
End Class
```

c) An application is designed to calculate area shape as follows. Fill in the blanks (a to h) following VB.NET code segment considering name of tools.

[Marks 04]

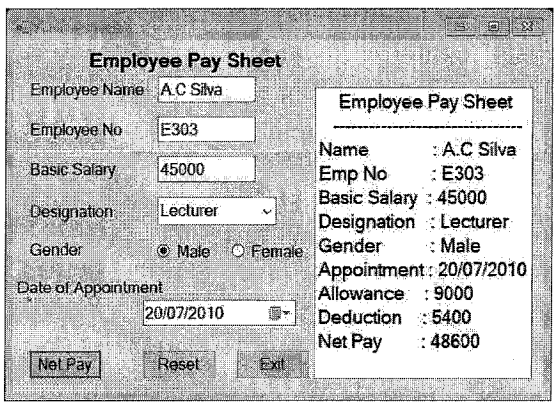



```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
btncal.Click
      Dim radius, width, length, height As Double
        Const pi As Double = 3.14
        Select Case ((a) txtshape.Text)
            Case 1
                radius = InputBox("Enter a Radius of Circle")
                lblname.Text = "Circle"
                lblValue.Text = pi * (b) radius ^ 2
            Case 2
                length = InputBox("Enter a Length of Rectangle")
                (c) width = InputBox("Enter a Width of Rectangle")
                lblname.Text = (d) "Rectangle"
                (e) lblValue.Text = length * width
            Case 3
                radius = InputBox("Enter a Radius of Cylinder")
                height = (f) InputBox("Enter a Height of Cylinder")
                lblname.Text = "Cylinder"
                lblValue.Text = pi * radius ^ 2 * (g) height

            (h) Case Else
                MsgBox("Invalid option")
        End Select
```

4)

a) An application is created in VB.NET to calculate pay sheet of a company.



Write VB.NET code segment to do the following as you enter "Net Pay" button using appropriate property name from the above table.

i. Display Name, No, Basic Salary, Designation, Gender and Date of Appointment of an employee in the listbox. [Marks 08]

ii. Display allowance of the staff according to the following scenario. [Marks 02]

Basic Salary	Allowance
Basic Salary < 40000	15 % of Basic Salary
Basic Salary >= 40000	20 % of Basic Salary

iii. Display deduction of the employee salary using following formula. [Marks 01]
*Deduction = (Basic Salary + allowance) * 10%*

iv. Generate Net Pof the staff using following equation. [Marks 02]
Net Pay=(Basic Salary + allowance) – Deduction

```
Private Sub BtnNetpay_Click(sender As Object, e As EventArgs) Handles
btnNetpay.Click
    Dim gen As String
    Dim net, dec, all As Double
   txtDetail.text="    Employee Pay Sheet " & vbcrlf &
  txtDetail.text= txtDetail.text & "   ------------------")& vbcrlf &
  txtDetail.text= txtDetail.text & "Name    : " & txtName.Text & vbcrlf &
  txtDetail.text= txtDetail.text & "Emp No  : " & txtEmpNo.Text& vbcrlf &
  txtDetail.text= txtDetail.text & ("Basic Salary : " & txtBsal.Text& vbcrlf &
txtDetail.text= txtDetail.text & "Designation: " & cmbDesig.SelectedItem&
vbcrlf &

        If (radMale.Checked = True) Then
            gen = "Male"
        ElseIf (radFemale.Checked = True) Then
            gen = "Female"
        Else
            gen = ""
        End If

           txtDetail.text= txtDetail.text & "Gender          : " & gen

        If (txtBsal.Text < 40000) Then
            all = txtBsal.Text * 0.15
        Else
            all = txtBsal.Text * 0.2
        End If
        dec = (all + txtBsal.Text) * 0.1
        net = (all + txtBsal.Text) - dec
        txtDetail.text= txtDetail.text & "Appointment : " & dtpDate.Text&
vbcrlf &   [
      txtDetail.text= txtDetail.text & "Allowance    : " & all & vbcrlf &
```

```
        txtDetail.text= txtDetail.text & "Deduction    : " & dec & vbcrlf &
```
[Marks 01]
```
        txtDetail.text= txtDetail.text & "Net Pay      : " & net & vbcrlf &
```
[Marks 02]
```
    End Sub
```

v. Write VB.Net code segment to clear all textboxes and list box while pressing "Reset" Button.

```
txtName.clear()
txtEmpNo.Clear()
txtBsal.Clear()
lstDetails.Items.clear()
```
[Marks 02]

vi. Give the code of the "Exit" button. [Marks 01]

```
Me.close()
```

b) Provide output of following coding [Marks 04]

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim price, qty As Integer
    price = 150
    qty = 20
    cal(price, qty)
    buy(price, qty)
End Sub

Public Sub cal(ByRef a As Integer, ByRef b As Integer)
    a = 100
    Dim amount As Integer = a * b
    MsgBox(amount)
End Sub

Public Sub buy(ByVal a As Integer, ByVal b As Integer)
    Dim amount As Integer = a * b
    MsgBox(amount)
End Sub
```

2000
2000

5)

a) Briefly explain namespace. [Marks 02]

Namespaces are file cabinets for classes, similar to the concept of folders. It can contain classes, other namespaces, which help to organise the vast number of classes in a neat structure.
To make use of a class, we need to reference or import the namespace that contains the class.

b) Object oriented approach is considered to be a good approach for developing program. State reason for it. [Marks 03]

Object oriented approach help to modeling real world object to solve complex problem through encapsulation.

Support complex data structure.
Reuse code through object library.
Greater semantic expressiveness.
Reduce procedurality through an enlarged vocabulary

c) Compare overloading and overriding methods. [Marks 03]

Overloading: Methods having identical names but different implementations
Several argument lists for calling the method.
Example: *MessageBox.Show* method
Overriding: Refers to a class that has the same method name as its base class.
Method in subclass takes precedence.

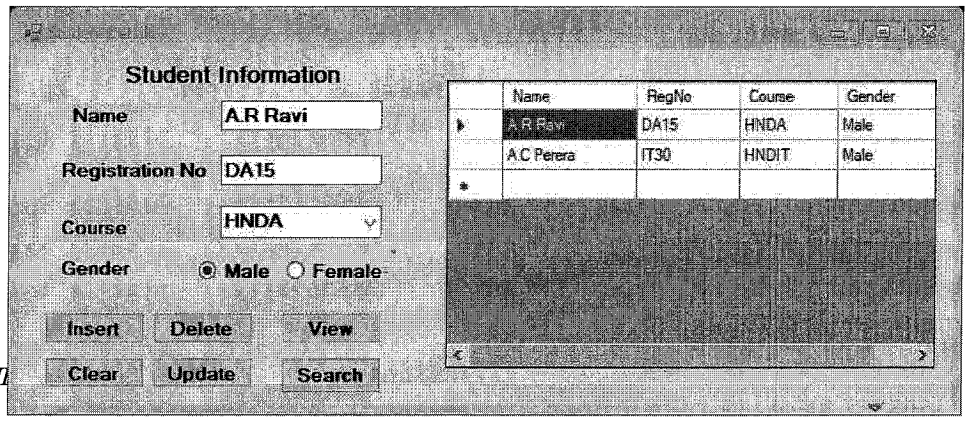
d) Create a code segment for following with object oriented concept [Marks 12]

i. Create a class named "Book".
ii. Declare three variable for name, price and quantity of book.
iii. Initialize these variable using parameterized constructor
iv. Generate function named "calAmount ()" which returns product of price and quantity.
v. Create object named "book1".
vi. Display amount in a message box by calling the function.

```
Public Class Book                                    [Marks 01]
    Public bname As String
    Public price As Double                           [Marks 02]
    Public qty As Integer
    Public Sub New(name, p, q)                       [Marks 02]
        bname = name
        price = p
        qty = q
    End Sub

    Function calAmount()                             [Marks 03]
        Return price * qty
    End Function
End Class
    Dim book1 As New Book("RAD", 400, 20)            [Marks 02]
    MsgBox(book1.calAmount())                        [Marks 02]
```

6) The following interface is created to store students' details in a database.



a) Give suitable name for user input control tools in the above interface according to the .NET convention. [Marks 02]

txtname, txtregno, cmbcourse, radMale, radFemale

*consider other database(SqlServer/Access) connections also

b) Provide a code segment for creating database connection to the above form with MySql database.

Hint: server=localhost, user id=root, password=Abc123, database=student table=details [Marks 02]

```
Imports mysql.Data.MysqlClient
Public Class Form1
    Dim con As MySqlConnection
    Dim cmd As MySqlCommand
   con = New MySqlConnection("server=localhost; user id=root;
   password=Abc123; database=student")
```

c) Write a code segment to insert a record of student into the database. [Marks 03]

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
     Dim gender As String
     If (radMale.Checked = True) Then
         gender = "Male"
     ElseIf (radFemale.Checked = True) Then
         gender = "Famale"
     End If
     con.Open()
cmd = New MySqlCommand("insert into details values('" & txtname.Text &
"','" & txtregno.Text & "','" & cmbcourse.SelectedItem & "','" & gender &
"')", con)
     cmd.ExecuteNonQuery()
     MsgBox("Record is inserted")
     con.Close()
End Sub
```

d) Create a code to update course of student whose registration number is "DA15". [Marks 03]

```
con.Open()
cmd = New MySqlCommand("update details SET  Name='" & txtname.Text & "','"
& txtregno.Text & "','" & Coursr=cmbcourse.SelectedItem & "', Gender=
gender where RegNo=DA15)", con)
     cmd.ExecuteNonQuery()
     con.Close()
```

e) Write a code segment to search details of student with registration number "IT30".[Marks 03]

```
Private Sub btnsearch_Click(sender As Object, e As EventArgs) Handles
 Dim dReader As MySqlDataReader
 Dim query As String = "select * from details  where RegNo = IT30" "'"
    cmd = New MySqlCommand(query, con)
    con.Open()
    dReader = cmd.ExecuteReader
```

```
        dReader.Read()
        If (dReader.HasRows = True) Then
            txtname.Text = dReader(0)

           Dim cor As String = dReader(2)
            cmbcourse.Text = cor

            Dim gender As String
            gender = dReader(3)

            If (gender = "Male") Then
                radMale.Checked = True
            ElseIf gender = "Female" Then
                radFemale.Checked = True
            End If
        Else
            MsgBox("Record is not available")
        End If
        con.Close()
    End Sub
```

f) Provide a code segment to view "details" table in a DataGridView. [Marks 03]

```
    Private Sub btnview_Click(sender As Object, e As EventArgs) Handles
btnview.Click
        Dim dAdapter As MySqlDataAdapter
        Dim dtable As New DataTable

        con.Open()
        dAdapter = New MySqlDataAdapter("select * from details, con)
        dAdapter.Fill(dtable)
        DataGridView1.DataSource = dtable.DefaultView
    End Sub
```

g) Write a code segment to delete a record of student whose registration number is "DA15"

```
    Private Sub btndelete_Click(sender As Object, e As EventArgs) Handles
    btndelete.Click
        con.Open()
 cmd = New MySqlCommand("delete from details where RegNo=DA15", con)
        cmd.ExecuteNonQuery()
        MsgBox("Record is deleted")
        con.Close()
    End Sub
```

[Marks 02]

h) Create a code segment to reset form by clicking clear button [Marks 02]

```
    Private Sub btnclear_Click(sender As Object, e As EventArgs) Handles
    btnclear.Click
            txtname.Clear()
            txtregno.Clear()
```

```
        cmbcourse.Text = "     "
        radFemale.Checked = False
        radMale.Checked = False
    End Sub
```