2.1 Write a function WordCount which accepts a file name as a parameter and returns an integer.

```
Imports System.IO

Imports System.Text.RegularExpressions

   Dim fileReader As String = File.ReadAllText("C:\VB\words.txt")

   Public Function WordCount(fileReader)

      Dim tCount As Integer

      If fileReader.Length = 0 Then

         tCount = 0

        Else

         tCount = fileReader.Split().length

      End If

      Return tCount

   End Function
```

2.2   Write a function ApperanceOfWord which accepts a file name and a word as parameters and returns an integer.

```
Imports System.IO

Imports System.Text.RegularExpressions

   Dim fileReader As String = File.ReadAllText("C:\VB\words.txt")

   Public Function ApperanceOfWord(fileReader, wordSearch)

      Dim wCount As Integer = Regex.Matches(fileReader, wordSearch).Count

      Return wCount

   End Function
```

2.3 Use the functions in 1 and 2 to write the FrequencyOfWord function.

```
Imports System.IO

Imports System.Text.RegularExpressions

   Dim fileReader As String = File.ReadAllText("C:\VB\words.txt")

   Public Function FrequencyOfWords(fileReader, wordSearch)

      Dim wCount As Integer = ApperanceOfWord(fileReader, wordSearch)

      Dim tCount As Integer = WordCount(fileReader)

      Dim total As Double = wCount / tCount

      Return total

   End Function
```

3.1 Explain the concept of a base class

- A base class is the class that other classes inherit from.
- All the properties and methods can be inherited by the derived class.

3.2 What is the difference between the class construct and structure construct [3]

- A Structure does not require a constructor, a Class does.
- Structures can have nonshared constructors only if they take parameters.
- Classes can have them with or without parameters.

3.3 provide a function calculateInterest which calculate the interest on the different accounts

```
Public MustInherit Class Account

  Protected _accType As String

  Public ReadOnly Property AccountType() As String

    Get
      Return _accType
    End Get
  End Property

  Public MustOverride Function calcInterest(amount As Double) As Double

End Class
```

3.4 Define a class savings account that is derived from the abstract account class. Define the calculate interest function

```
 Class SavingsAccount

  Inherits Account

  Public Overrides Function calcInterest(ByVal amount As Double) As Double

    Dim interest As Double

    Select Case amount

      Case Is < 1000

        interest = amount * 0.1

      Case Is < 10000

        interest = amount * 0.12

      Case Is > 10000

        interest = amount * 0.135

    End Select

    Return interest

  End Function

End Class
```

4.1 In your explanation list the main dialogs and options that the programmer will make use of. [5]

- **Choose a Data Source Type**

  Select the type of data source to create from the available options listed.

- **Choose a Database model**

  You can choose Dataset or Entity Framework

- **Choose the data connection.**

  You can use an existing one or create a new one.

- **Choose if you want to save the connection string in the application configuration file (App.config).**

  It is better if you do because then you can reference the connection by name.

- **Choose database objects you want to include.**

  Choose from any tables, views or functions stored in dataset.

4.2 Describe the main steps in using a DataGridView controll which is part of ADO .NET [5]

- Drag table from data sources onto the form
- VS creates controls:
  - DataSet object
  - TableAdaptorManager object
  - BindingSource object
  - BindingNvigator object
- Binds controls to data source to:
  - Add
  - Update
  - Delete

List five possible situations needing error handling in a program. Briefly explain each problem

- **Format Exception**
  Represents an exception that occurs when the format of an argument doesn't meet the parameter specifications of the invoked method.
- **Arithmetic Exception**
  Represents an exception that occurs during an arithmetic, casting, or conversion operation.
- **Overflow Exception**
  Occurs when the result of an arithmetic operation is too large for the receiving variable.
- **A divide-by-zero exception**
  Occurs if an application attempts to divide a number by zero.
- **Invalid Cast Exception**
  It represents an exception that occurs during explicit casts with Visual Basic functions like CDec.

Explain the difference between a module and a form, and also explain the purpose of each construct

- Module
  A module typically contains procedures, variable, and constants that are used by more than one form.
- Form provides a use interface that let the user interact with the application