Basic Hardware Concepts

Programming Paradigms
   Procedural
   Object Oriented
   Event Driven

What is VB.Net?
   Windows Applications (Object Oriented, Event Driven)
   Console Applications (Procedural or Object Oriented)

The Object Model
   Class
   Objects (Built-in and user-defined)
   Properties
   Methods
   Events

Steps in Writing a Typical VB project.
   Planning: (GUI, properties, pseudocoding)
   Coding: (Convert the GUI to Forms and Controls, Set the properties, Convert
            the Pseudocode to VB code, Test and Debug.)

The Software Development Life Cycle (SDLC)
   Planning
   Analysis
   Design
   Implementation
   Testing
   Maintenance

Compile (syntax vs. run-time vs. logical errors)

Variables (represents memory, has a type and size)
   DIM strName as string
   integer, double, decimal, boolean, char, byte, string, etc..

   Global vs. Local
   Why initialize variables?

Constants
CONST TAX_RATE as Decimal = 0.08

Variable and Constant Scope
1) Module level (within the form)
2) Local level (with a procedure

Option Explicit ON (variables cannot be used without being declared first. ON by default)
(Turn it OFF if you have old VB programs that you are trying to compile and run quickly.) (Should be left ON for Safety)

Option Strict ON (Makes VB a strongly typed language, No automatic type conversion. Must use the type conversion functions)

Type Conversion functions
CInt(x)
CStr(x)
CDec(x)
CLng(x)
Cdbl()

GUI Components:
- Forms, Label, Textbox, Checkbox, Button, RadioButton, ListBox, Combobox, PictureBox, GroupBox, DataGrid.
- Tool Tips and Component Trays.
- Setting the focus i.e. txtName.focus()
- Setting up Buttons with Keyboard Access Keys. (btnOK.text = &OK)
- Setting up a default button for a form (Form.AcceptButton = btnOK)
- Setting up a Cancel button for a form (Form.CancelButton = btnCancel)

Concatenation and Continuation: (& and _)

Arithmetic operators (+, -, /, \\, *, MOD, ^)

Relational operators ( =, <=, >=, <, >)

Input and output (=, <=, >=, <, >)

Input and output
Console Mode: Console.ReadLine(), Console.WriteLine()
Windows Applications: MessageBox.Show(), InputBox()

Branching:
  Using the IF Statement:
  (If, If-then-else, nested if statements)

  Using the (Select Case) statement:
  Select Case Expression
case X
    Code to run
case Y
    Code to run
case else
    Default case
End select

Loops
    (for.. next, do while ...Loop, do Until ... Loop)
    Necessary conditions for a loop (how to get in, and how to get out)

Truth Table

Problem Solving Methodology
    Top down design
    Break the problem into smaller, more manageable tasks.
    Divide and conquer
    Encourages modular design
    Defers the details till later

Functions and Procedures

Procedures and Functions:
    Passing arguments
        (Pass by value vs. Pass by reference, when?, why?)
    Formal vs. actual parameters
    Returning values from functions
        Via the return statement vs. the function’s parameter list.

Arrays
    One and Two Dimensional
    Uses of arrays
    Operations on the Array:
        Array as an Abstract Data Type (ADT)
        Initialize, load, print, search, sort, etc...

Structures
    Array of Structures
    Arrays as elements of Structures (Redim)

Classes and Objects
    Encapsulation, Inheritance, Polymorphism
    Creating new classes
    Private vs. Public class variables
    Private vs. Public methods
    Instantiating objects
Class Constructor
Overloading
Overriding

File Concepts and operations
 imports system.io
The stream concept
Reading, Writing
XML format and concepts

On Your Own Reading

Robust I/O and Input validation:
 IsNumeric()

Formatting Functions:
 $12 = FormatCurrency(12)
 5% = FormatPercent(0.05)

String Manipulation:
 String Length, TrimStart(), TrimEnd(), Trim(), Remove(), StartsWith(), EndWith(),
 ToUpper(), ToLower(), SubString Manipulation, Replace(), Mid(), PadLeft(),
 PadRight(), Insert(), IndexOf(), the Like Operator.

Advanced Concepts (Optional)

Database Concepts
 Basic DB concepts: (Database, file, record, field, meta data, DBMS, SQL, QBE)
 VB concepts and objects for accessing databases. (ADO.Net, DataAdapter,
 Connection, DataSet objects, SQL statements.

Binding VB controls to Database Fields
 Textbox, Comboboxes, Label, DataGrid

Dynamic (programmatic) allocation of controls
 Such as (Textboxes, CheckBoxes, RadioButtons, etc.)

Event Handling
 Sharing event handlers (Handling multiple events via the same event procedure)
 Using CType() function to convert generic senders to