Basic Hardware Concepts
Programing 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)
Clng(x)
Cdbl()

GUI Components:
- Forms, Label, Textbox, Checkbox, Button, RadioButton, ListBox, Combobox, PictureBox, GroupBox.
- 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)
- Tool Tips and Component Trays.
- Setting the focus i.e. txtName.focus()

Concatenation and Continuation: (& and _)

Arithmetic operators (+, -, *, /, \

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

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

Input and output
Console.ReadLine()
Console.WriteLine()
MessageBox.Show()
InputBox()

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

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

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
Passing arguments (Pass by value vs. Pass by reference, when?, why?)
Formal vs. actual parameters
Returning values from functions via the return statement.

Truth Table