Basic Hardware Concepts
    (ROM, RAM, CPU, ALU, CU, Input/Output/Storage/Network Devices)

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

Steps in Writing a Typical VB project.
    Planning:  (GUI, properties, pseudocode or flowchart)
    Coding/Debugging:  (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

Errors (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 (variable declared within the scope of class or module)
    2) Local level (variable declared within a procedure)

Compiler Directives
    Option Explicit ON  (Variables cannot be used without being declared first.)
    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

Concatenation and Continuation characters:  (& and _)

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

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

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

<table>
<thead>
<tr>
<th>OR</th>
<th>AND</th>
<th>NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>T OR T = T</td>
<td>T AND T = T</td>
<td>NOT T = F</td>
</tr>
<tr>
<td>T OR F = T</td>
<td>T AND F = F</td>
<td>NOT F = T</td>
</tr>
<tr>
<td>F OR T = T</td>
<td>F AND T = F</td>
<td></td>
</tr>
<tr>
<td>F OR F = F</td>
<td>F AND F = F</td>
<td></td>
</tr>
</tbody>
</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.
Event Procedures (hint: look for “Handles” keyword at the end of procedure declaration)

Arrays
Homogeneous aggregate of data elements
Creating many variables and accessing them via one variable name
Uses of arrays
Operations on the Array:
  Array as an Abstract Data Type (ADT)
  Initialize, load, print, search, sort, etc...

Structures
Structures provide a heterogeneous aggregate of data elements

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

What are the ramifications of designing “BAD” systems or writing “BAD” code, or designing “BAD” interfaces?
on other humans
on the society
**On Your Own Reading**

**Robust I/O and Input validation:**
- IsNumeric()

**Formatting Functions:**
- $12 = \text{FormatCurrency}(12)
- 5% = \text{FormatPercent}(0.05)

**String Manipulation:**
- String Length, TrimStart(), TrimEnd(), Trim(), Remove(), StartsWith(), EndsWith(), 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