Review for Test 1	I101 / B100	Hossein Hakimzadeh

- Basic Hardware Concepts
  - memory RAM/ROM, processor, CPU, ALC, CU, BUS, Input devices, output devices, storage devices
  - Binary (0, 1), ASCII code
- Programming Paradigms
  - Procedural
  - ~~Object Oriented~~
  - Event Driven
- What is VB.Net?
  - Windows Applications (Object Oriented, Event Driven)
  - Console Applications (Procedural or Object Oriented)
- 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
- Errors (syntax vs. run-time vs. logical errors)
- Variables (represents memory/RAM, 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~~
- What is a compiler?
- 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)
  - Events and event handlers
- Random number generation: Randomize(), Rnd()
- ~~Concatenation and Continuation: (& and _)~~
- Arithmetic operators (+ , - , / , \, * , MOD, ^)
- Relational operators (= , <= , >= , <>)
- 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)
- 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 (AND, OR, NOT)
  - Evaluating Boolean expressions
- Writing algorithms, writing pseudocode, drawing flow chart
- Identifying Syntax errors
- Code walk through (given an algorithm or code segment, how do we disk check the code.)