I101/B100
Problem Solving with Computers

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend
What do we need to learn in order to write computer programs?

- Fundamental programming constructs:
  - Variables,
  - Arithmetic operators,
  - Input and output
  - Conditionals,
  - Loops,
  - Procedures and functions,
  - Arrays,
  - Structures, classes and objects,
  - Files
Problem Solving Methodology

- *How do software engineers solve large problems?*

- *How does anyone solve a large problem?*

- *What is a large and complex problem?*
  - Building a car?
  - Building a house?
  - Building a word processor?
  - Planning a wedding?
  - Completing a degree?
Problem Solving Methodology

How does anyone solve a large problem?

In order to solve large problems we need to be able to **break them down** into **smaller and more manageable sub-problems**.
Top-Down Problem Solving Strategy:

- Breaks the problem into smaller and more manageable tasks.
Top-Down Problem Solving Strategy:

- Divide and Conquer strategy
- Breaks the problem into smaller and more manageable tasks.
- Encourages modular and structured programming. One entry point one exit point.
- Defer the details as long as possible.
Top-Down Problem Solving Strategy:

Program

- Payroll
  - Check Printing
  - Electronic Deposit
- Account Receivable
  - Invoice
- Account Payable
  - Check Printing
- Purchasing
  - Purchase Orders
Procedures and Functions

- A small module (block of code) designed to perform a specific task.

- Procedures and functions are the primary mechanism by which large programs can be subdivided into smaller and manageable components.

- Procedures and functions allow programmers to avoid duplicating code in different parts of the program.
How do we get the procedure or a function to do something?

- Think of a procedure or function as someone you contract to do a task for you.

- You call the contractor and ask them to do the job!
Sub-programs in VB .Net can be:

- **Procedures**
  - (Perform an assigned task)

- **Functions**
  - (Perform an assigned task, and then return a value)
Procedures

- **Sub Procedures in VB.Net:**

  1. **Event Procedures**
     - (A collection of code associated with a GUI control. Such a procedure is invoked when an event related to the GUI occurs.)

  2. **Independent Procedures**
     - (A collection of code that can be invoked by the programmer from one or more places in the program.)
Procedures

- Syntax:

  **Private Sub**  `procedureName([OptionalParameterList])`

  `[Statements]`

  **End Sub**
Procedures without Parameters:

- Example:

```vba
Private Sub DisplayMessage1()
    Console.WriteLine("Hello World")
End Sub
```

Note: This procedure has **no parameters**.
Procedures with Parameters:

Example:

Private Sub DisplayMessage2(ByVal Message As String)
    Console.WriteLine(Message)
End Sub

Note: This procedure has one Parameter (Pass By Value).
Calling a procedure:

Sub Main()
    DisplayMessage1()
    DisplayMessage2("Hello South Bend")
End Sub
A complete program

Option Explicit On
Option Strict On

Module Module1

Sub Main()
    DisplayMessage1() 'No Arguments
    DisplayMessage2("Hello South Bend") 'Pass by Value
    DisplayMessage2("Hello Mishawaka") 'Pass by Value
    Console.ReadLine()
End Sub

'Procedure without a Parameter
Private Sub DisplayMessage1()
    Console.WriteLine("Hello World")
End Sub

'Procedure with Parameter (Pass By Value)
Private Sub DisplayMessage2(ByVal Message As String)
    Console.WriteLine(Message)
End Sub

End Module