Visual Programming
C#
Week 2, 3

Tahir Nawaz

What is C#?

- C# (pronounced "C sharp") is an object-oriented language that is used to build applications for the Microsoft .NET platform
- C# is designed by Microsoft to combine the power of C/C++, Java and the productivity of Visual Basic
- The goal of C# and the .NET platform is to shorten development time
  - By allowing developers to spend their time working on the application logic instead of low level programming details

Compiling C# Source Code

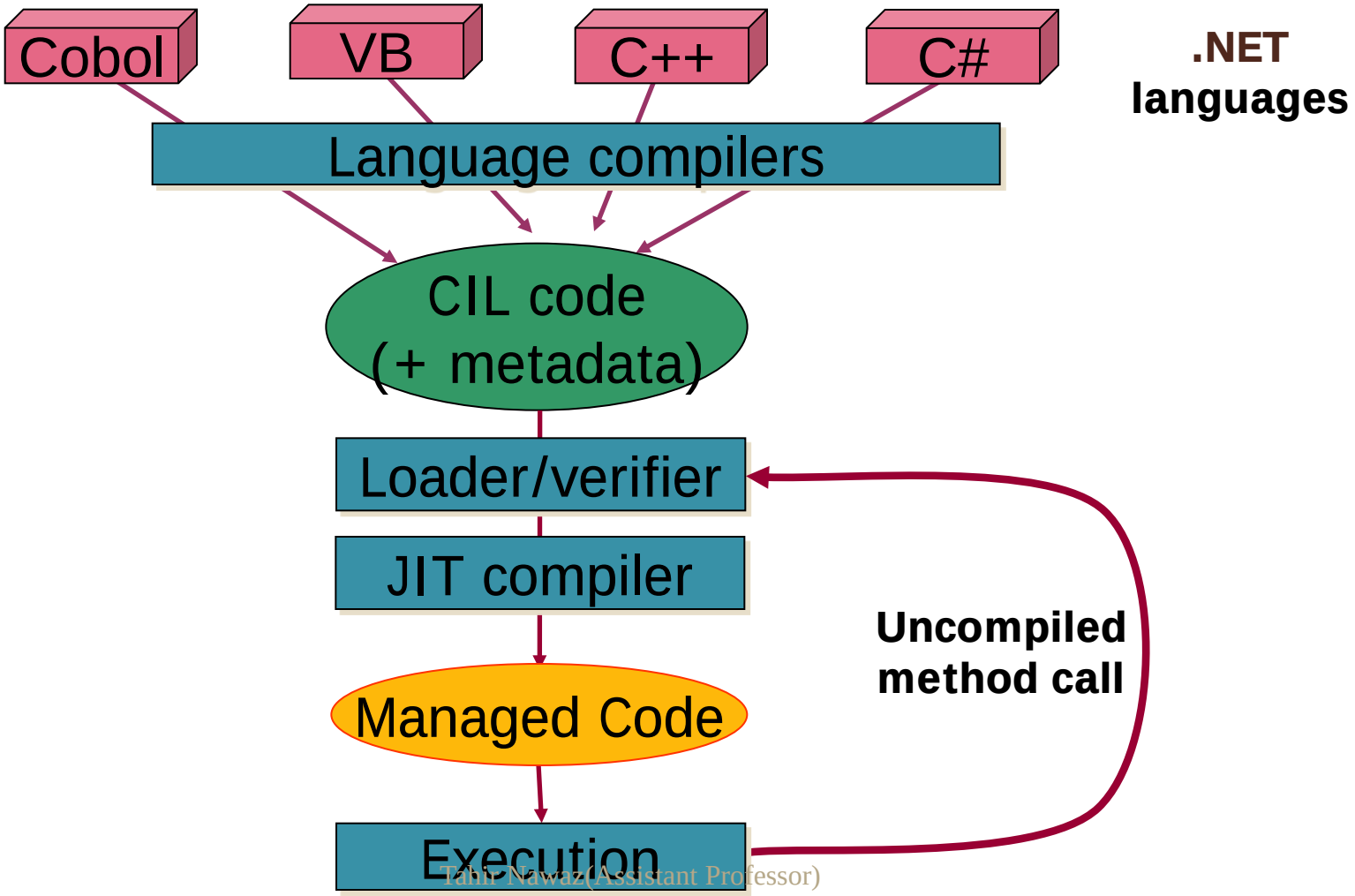
- C# file names have the extension .cs
- To create the IL file, we need to compile the .cs file using the csc.exe (using the command line), followed by the name of the source file
- The result is a file with the same name but the .exe extension, which is called an assembly
- The assembly file contains all the information that the common runtime needs to know to execute the program

e.g. Csc.exe File.cs

Features of C#

- C# syntax is very similar to Java (and thus C++)
- C# features are very similar to Java
  - Object-orientation with single inheritance
  - Support of interfaces
  - No pointers (except for unsafe code)
  - Exceptions
  - Threads
  - Namespaces (like Packages)
  - Strong typing
  - Garbage Collection
  - Reflection (In the .NET universe, reflection is the process of runtime type discovery)
  - Dynamic loading of code

Execution model C# and other Languages



Namespaces

- A C# namespace is a way to group classes and is used in a manner similar to Java's package construct
- C# namespaces are similar to C++ namespaces and syntax
- Unlike Java, C# namespaces do not dictate the directory structure of source files in an application
- Namespaces can be nested similar to Java packages

.NET Framework Class Library

- The .NET Framework Class Library is made up of various namespaces. *Namespaces* are actually collections of types, logically organized. This enables you to have multiple versions of types with the same name but in different namespaces, thereby avoiding conflicts.
- Just as with a library you have collected in your home of useful books, the class library is a set of types that not only make up the .NET Framework itself, but also are available to developers for their use.
- Another big benefit of using the .NET Framework Class Library is to be able to use the classes in your applications consistently no matter whether you are using C# or Visual Basic .NET, Windows, or Web forms. Namespaces can also contain other namespaces in a hierarchical way. By creating sub-namespaces, you can categorize your types for use at different times.

.NET Framework Class Library

- When you create a .NET application, C# Express creates references to different namespaces, based on what kind of project you are creating.
- The best way to understand the .NET Framework Class Library is to take a look at some of the namespaces in it.

Namespace	Description
`System`	Main system namespace that is broken into many categories.
`System.Data`	Makes up the classes used for ADO.NET, and overall data manipulation of just about any kind. Sub-namespaces of the `System.Data` include `System.Data.SqlClient` and `System.Data.OleDB`.
`System.Drawing`	Used for drawing shapes and objects in your applications.
`System.Windows.Forms`	Namespaces and classes for creating Windows forms applications.

Object Browser: Tool of the Namespace Trade

- One of the tools worthwhile to look at when you are learning about namespaces is the Object Browser.
- With the Object Browser we can search through and locate the syntax for various classes you want to use. Once you locate the class in the namespace, you can press F1 and get help on it if necessary.
- The Object Browser is a great tool when you just want to look through a namespace to get an idea of what is included.

The Using Directive

- When the code window of any blank or new project we have a space called region also known as code blocks at the top made of using directives by default each and every type of project we have a list of namespaces those are included in the project to facilitate the programmer from basic classes or collections also called namespaces.

The Using Directive

- using system;
- using System.Collections.Generic;
- using System.ComponentModel;
- using System.Data;
- using System.Drawing;
- using system.Text;
- using system.Windows.Forms;

The Differences between C# and C# Express

- C# Express is actually what is called an *IDE*, or *integrated development environment*
- Language we are reading this in is that C# Express is a set of tools, including a special text editor that enables you to write computer programs in C#, the software development language.
- It also handles other necessary tasks such as building your application to either test for errors

The Differences between C# and C# Express

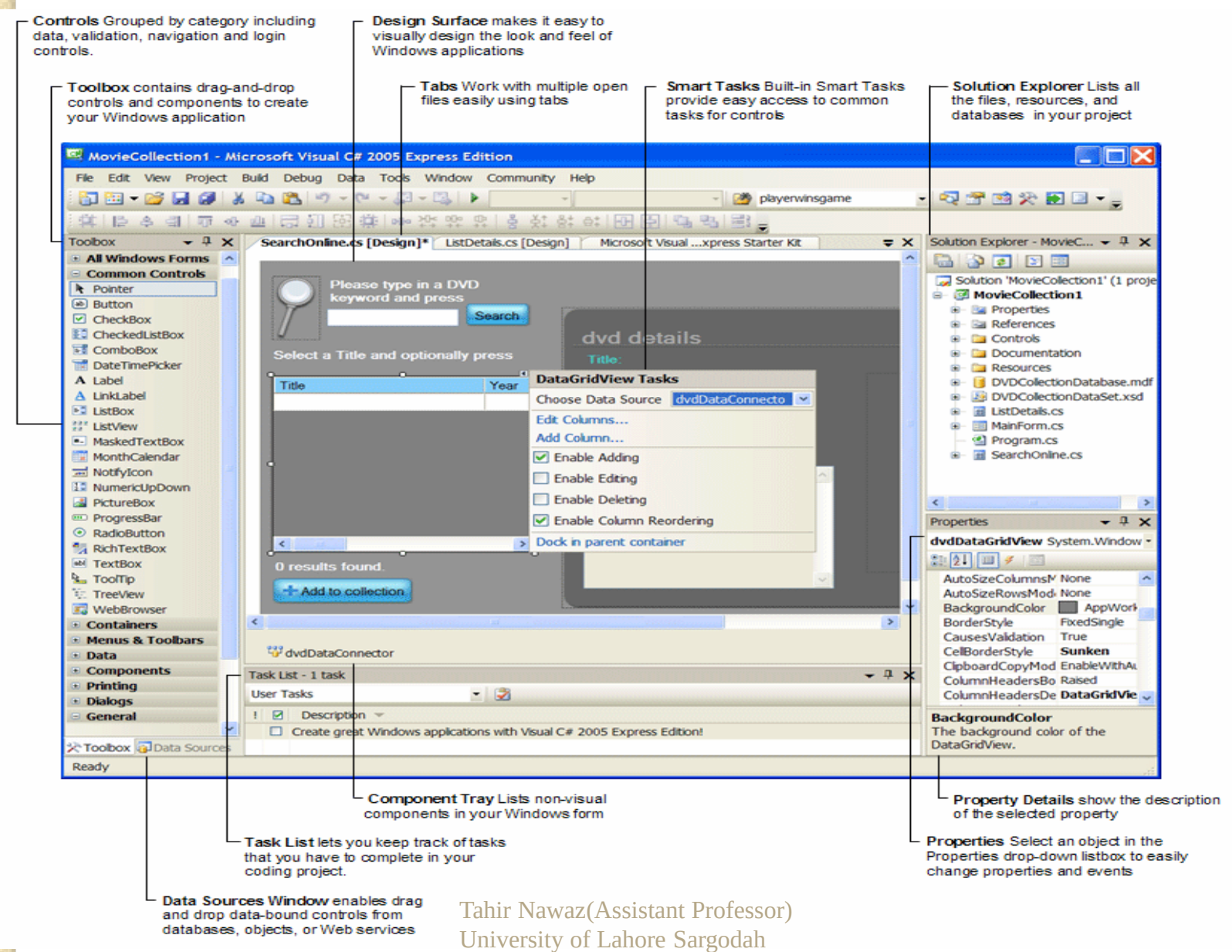
- Microsoft wanted to come up with a way to get those who are not yet C# developers and interested in programming of C#.
- There are a couple of different ways to create the C# applications.
  - **Use a simple note pad or third-party editor, and then use the command-line compiler.** It is Really, only long-time hardcore developers use this method, where we need no support for development and want to struggle through compiling the programs by themselves.
  - **Use of Visual Studio .NET to development and maintain your C# code and application.** This is the preferred method if you can afford Visual Studio.

Microsoft introduced the Express series to give you experience with developing using the last method but with pared-down features. The full-blown versions of Visual Studio contain supersets of commands found in the Express versions

Members of the Express Series

- In an effort to expose new developers of all kinds to their different products, Microsoft has created the Express series
- Besides C# Express, other products in the series include:
  - Visual Basic 2005 Express
  - Visual C++ 2005 Express
  - Visual J# 2005 Express
  - Web Dev 2005 Express
- The first three in this list are additional programming languages. Web Dev Express introduces you to Web development with ASP.NET and can be used with each of the four languages in the Express series.

C# IDE


Controls Grouped by category including data, validation, navigation and login controls.
Design Surface makes it easy to visually design the look and feel of Windows applications
Toolbox contains drag-and-drop controls and components to create your Windows application
Tabs Work with multiple open files easily using tabs
Smart Tasks Built-in Smart Tasks provide easy access to common tasks for controls
Solution Explorer Lists all the files, resources, and databases in your project
MovieCollection1 - Microsoft Visual C# 2005 Express Edition
File Edit View Project Build Debug Data Tools Window Community Help
playerwinsgame
Toolbox
All Windows Forms
Common Controls
Pointer
Button
CheckBox
CheckedListBox
ComboBox
DateTimePicker
Label
LinkLabel
ListBox
ListView
MaskedTextBox
MonthCalendar
NotifyIcon
NumericUpDown
PictureBox
ProgressBar
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
WebBrowser
Containers
Menus & Toolbars
Data
Components
Printing
Dialogs
General
SearchOnline.cs [Design]*
ListDetails.cs [Design]
Microsoft Visual ...xpress Starter Kit
Please type in a DVD keyword and press
Search
Select a Title and optionally press
dvd details
Title:
Title
Year
DataGridView Tasks
Choose Data Source
dvdDataConnecto
Edit Columns...
Add Column...
Enable Adding
Enable Editing
Enable Deleting
Enable Column Reordering
Dock in parent container
0 results found.
Add to collection
dvdDataConnector
Task List - 1 task
User Tasks
Description
Create great Windows applications with Visual C# 2005 Express Edition!
Toolbox
Data Sources
Ready
Solution Explorer - MovieC...
Solution 'MovieCollection1' (1 proje
MovieCollection1
Properties
References
Controls
Documentation
Resources
DVDCollectionDatabase.mdf
DVDCollectionDataSet.xsd
ListDetails.cs
MainForm.cs
Program.cs
SearchOnline.cs
Properties
dvdDataGridView System.Window
AutoSizeColumnsM None
AutoSizeRowsMod None
BackgroundColor
BorderStyle FixedSingle
CausesValidation True
CellBorderStyle Sunken
ClipboardCopyMod EnableWithAu
ColumnHeadersBo Raised
ColumnHeadersDe DataGridVie
BackgroundColor
The background color of the DataGridView.
Component Tray Lists non-visual components in your Windows form
Task List lets you keep track of tasks that you have to complete in your coding project.
Data Sources Window enables drag and drop data-bound controls from databases, objects, or Web services
Property Details show the description of the selected property
Properties Select an object in the Properties drop-down listbox to easily change properties and events

Steps to open console application

1. Choose Program Files → Visual C# 2005 Express from the Windows Start menu. The IDE opens, and the Start Page is displayed.

2. Select New → Project from the Files menu. The New Project dialog box appears, giving you a choice of templates.

3. Highlight Console Application. Remember that this type of application doesn't have any forms or interface. It is also the easiest to start with.

4. Type in the name of the project you want to create. For this Try It Out, Chapter1Console was used, as shown in Figure 1.

5. Click OK. Your project is now created, as shown in Figure 2.

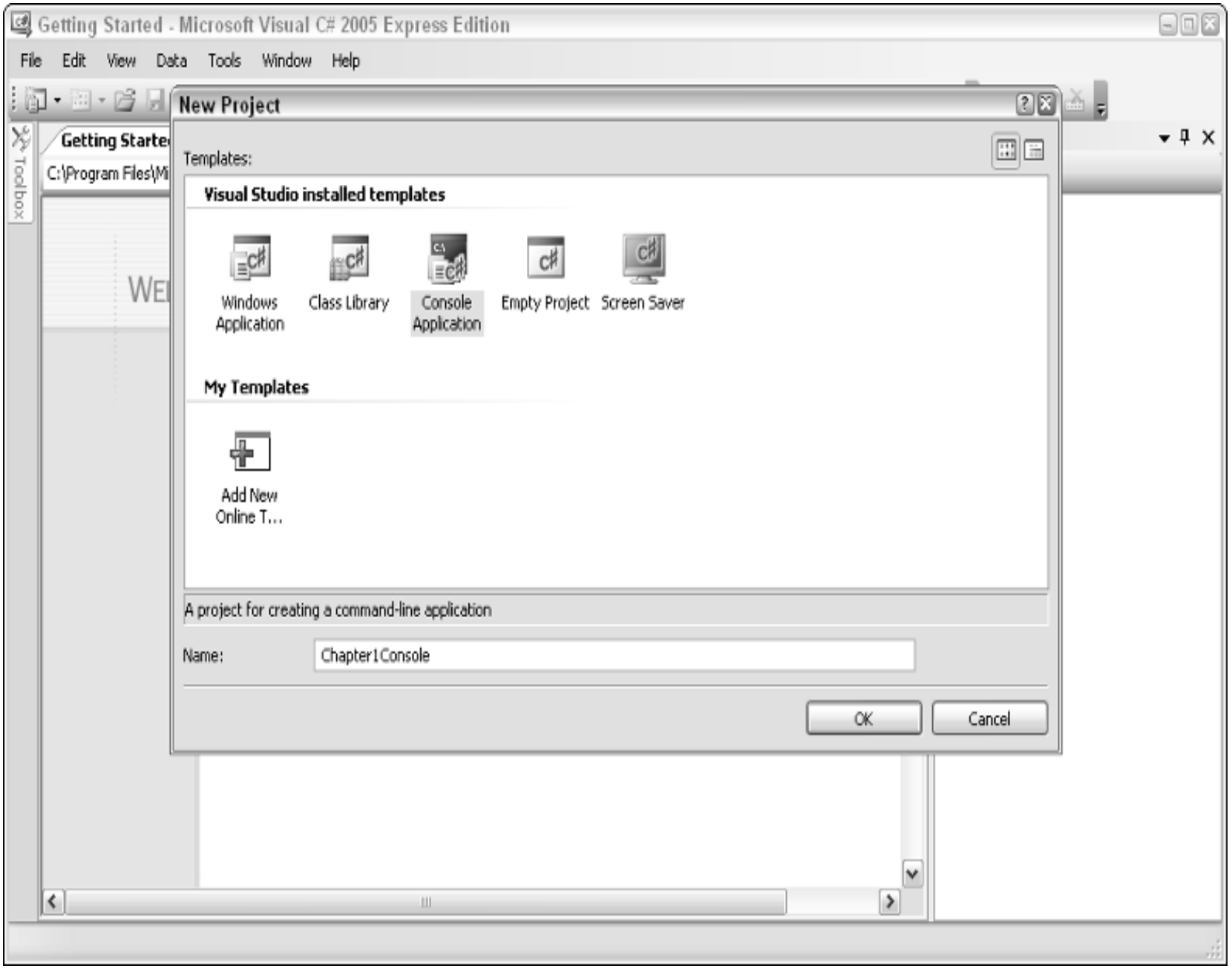

Getting Started - Microsoft Visual C# 2005 Express Edition
File Edit View Data Tools Window Help
New Project
Templates:
Visual Studio installed templates
Windows Application
Class Library
Console Application
Empty Project
Screen Saver
My Templates
Add New Online T...
A project for creating a command-line application
Name:
Chapter1Console
OK
Cancel
Toolbox
Getting Starte
C:\Program Files\Mi

Figure 1

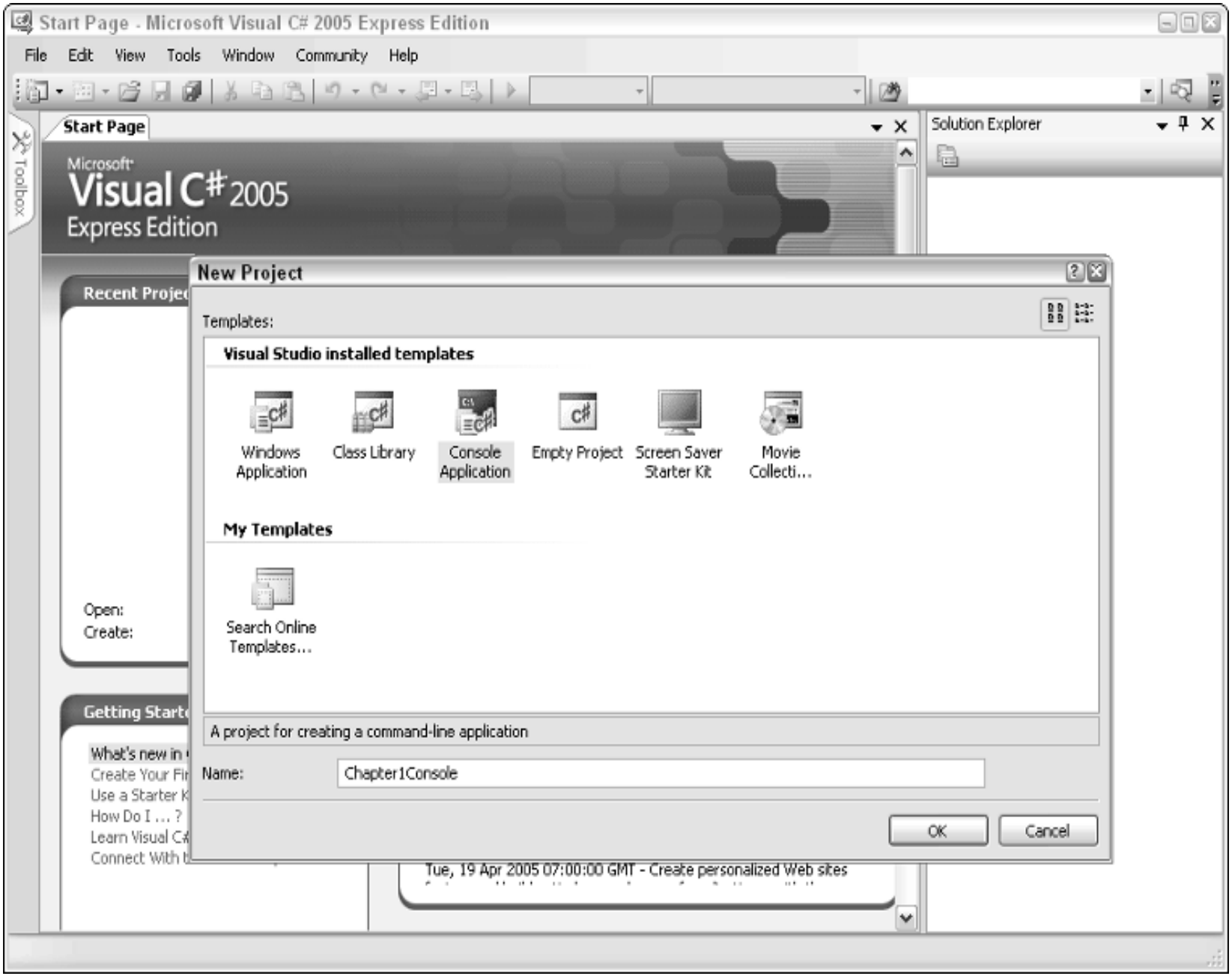

Start Page - Microsoft Visual C# 2005 Express Edition
File Edit View Tools Window Community Help
Start Page
Toolbox
Microsoft
Visual C# 2005
Express Edition
Solution Explorer
New Project
Templates:
Visual Studio installed templates
Windows Application
Class Library
Console Application
Empty Project
Screen Saver Starter Kit
Movie Collecti...
My Templates
Search Online Templates...
A project for creating a command-line application
Name:
Chapter1Console
OK
Cancel
Recent Proje
Open:
Create:
Getting Start
What's new in
Create Your Fir
Use a Starter K
How Do I ... ?
Learn Visual C#
Connect With t
Tue, 19 Apr 2005 07:00:00 GMT - Create personalized Web sites

Figure 2

Data Types

Type	.NET Data Type	Description
bool	System.Boolean	True/false values
byte	System.Byte	1 or 0 values
char	System.Char	Single characters
decimal	System.Decimal	Decimal values
double	System.Double	Large floating-point number
float	System.Single	Small floating-point number
int	System.Int32	Integer value
long	System.Int64	Large integer value
object	System.Object	Generic object that can hold other types
short	System.Int16	Short integer value
string	System.String	Array of characters

Constants

- Constants are different from variables in that you will declare and assign their values once in a module or namespace.
- They are useful when you use a value that may mean little when viewed as a number but makes perfect sense as a label.
- Create a constant by using the const statement

Naming C# Variables

- some people tend to use the shortest variable names they can. For e.g, if they are declaring a variable that stores the last name of a person, they would type string ln;
- A number of standards are used for naming variables in C# and other programming languages. A couple
- of common standards used for .NET development are as follows

- **Camel notation.** Takes the first word and displays it using lowercase and then uses proper case on the second part of the variable name. suppose we have a variable that stores a person's
- last name. You would declare and name the variable like this:
  - string lastName;
- **Hungarian notation.** In this naming standard, you place the type of data you are using for the
- variable as the prefix for the variable name. The rest of the name is then typed in proper case.
- This is the version I tend to use, because I like being able to see what type of data I am working
- with. With this notation, the previous example would look like this:
- string strLastName;

Enumerations

- *Enumerations* are a type of variable that can be used to reflect various values.
- Mainly when using a class such as MessageBox. When you call the Show method of the MessageBox class, you can pass an argument to the method. This is also very useful when you are passing an argument to your own procedure and you want to limit the values sent.
- For example, say you want to create an enumeration called intMonths. In this variable, you create enumerators for each month in a year. The declaration for such an enumeration would look like the following:

```
enum Months
{
        January, February, March,
        April, May, June, July, August,
        September, October, November, December
};
```