

1: C/C++ Application Learning Trail - NetBeans Tutorials, Guides and Articles

"Application Development Using C# www.enganhecubano.com" is an excellent introduction to www.enganhecubano.com framework. This book is for the experienced developer with knowledge of an object oriented language such as Java or C++.

Please Sign up or sign in to vote. If you love C and want to create an Android application using that language then you have to thank Xamarin as they created this great Cross Platform development tool which enables developers to develop iOS and Android apps in C language. I tried developing before in Android but I did not like the experience in using the available IDE, as its slow and resource hungry, so I gave up. Xamarin is offered in different licenses from free to enterprise levels but for not I will be using the starter version which is the free version. It includes the Xamarin Studio which is great start for those who want to try out creating their first apps for Android, they also offer a Business license which lets you develop in Visual Studio so you can use that rich experience similar to developing Web Apps or Windows Apps, then they have this Enterprise which contains everything. For now lets see what the free stuff can offer, I did a simple project which I will be discussing below and it looks very promising, I love it. Like what I said above we will be using the free version so we will make this simple project by using Xamarin Studio. Most of the calculator projects online regardless its C , VB. First you need to download Xamarin at <https://www.xamarin.com>: Fire up Xamarin Studio and select new Solution then Android Application, give it a solution name and in this case we use "Calculator". Once all ok you will see the IDE which is like a cross between Visual Studio and Eclipse, it has intellisense and fancy text colours and highlighting. At this point we are more interested on the 2 files on the solution, that layout and the MainActivity. First lets design you calculator, you need to go to that Main. To see how each of them would look like here is an illustration Linear Layout " you can organize objects in either horizontal or vertical rows. Relative Layout " specify the location of child objects relative to each other. For example object A is placed on the left of object B or it can be aligned to a parent. List View " Displays it in a scrolling single column list. Grid View " Displays it in a scrolling grid view of columns and rows. Now lets start designing, for this project we will using a mix of Linear and Relative. Copy and Paste the code below then I will give you some explanation of what it does.

2: Building an N-Tier Application in .NET

Application Development Using C# www.enganchecubano.com gives experienced developers unprecedented insight for www.enganchecubano.com enterprise applications with C#. A running case study covers the entire process: creating a monolithic C# console application; adding a Windows Forms interface; isolating functionality inside components, providing database access and.

Collapse the table of content Expand the table of content This documentation is archived and is not being maintained. This documentation is archived and is not being maintained. NET is a general purpose development platform. It can be used for any kind of app type or workload where general purpose solutions are used. It has several key features that are attractive to many developers, including automatic memory management and modern programming languages, that make it easier to efficiently build high-quality apps. NET are available, based on open. NET Standards that specify the fundamentals of the platform. NET Framework, which is one of the existing. This topic will also talk about other. NET technologies and where you can find their related documentation. It can be installed locally with your app with only the packages you need. It provides a lightweight development model and the flexibility to work with your favorite development tools on your favorite development platform. NET Core page to find installation instructions for each supported platform. You can currently use. NET Core to develop console or Web applications: To read about developing modern cloud-based Web application, see the ASP. To read about working with data, see the Entity Framework documentation. NET Framework, see the overview. We update the latest versions of the. NET Framework documentation on a regular basis with content fixes and enhancements, but we do not maintain older versions. Earlier versions of the. NET Framework documentation are available from the table of contents pane on the left. You can use the. To read about creating Windows 8. For information about creating portable. For additional information about developing apps, visit the Windows desktop apps and web development sections of the MSDN Library. You can use Visual Studio for your development tasks and select from a wide range of programming languages. NET Framework also releases out-of-band packages with new functionality and improved cross-platform support. For information about these, see The. You can extend the capabilities of your apps with the following.

3: Developing Universal Windows Apps with C# and XAML – Microsoft Virtual Academy

The release www.enganhecubano.com Core means that cross-platform development with C# is finally here. In this fast-paced practical course you'll work through a rapid introduction to C# followed up by an overview www.enganhecubano.com in the context of two real-world applications.

Estimated time to complete this tutorial 20 minutes
Tutorial Objective In this tutorial, you will learn how to leverage the design capabilities integrated into Visual Studio in order to rapidly develop applications that use data stored in a SQL Server Compact Edition database. You will be using Visual C. NET in this tutorial. There is another tutorial that shows how to use Visual Basic. NET to perform the same task. In this tutorial, you will perform the following exercises: In the Templates box, select Windows Application. The Data Source Configuration Wizard appears. Still on the Add Connection dialog box, click Browse to open the file-selection dialog box. Adding a connection Select the file C: On the Add Connection dialog box, click Test Connection to verify that the database connection is setup properly. If the database file exists and SQL Server Compact Edition has been properly installed, then a dialog box informs you that the connection succeeded. Click OK on the Add Connection dialog box, now that you have successfully tested the connection. You can now expand the Connection string on the Choose Your Data Connection dialog box to view the connection string that has been generated, as shown in Figure 6. A prompt appears asking whether or not you would like to copy the local data file into your project and modify the connection accordingly, as shown in Figure 7. Click Yes to copy the local data file into the project. Prompt regarding copying local data file to the current project Click on the picture for a larger image When you are prompted to save the connection string, be sure the Yes, save the connection as check box is checked. Leave the connection string named NorthwindConnectionString, and then click Next, as shown in Figure 8. Visual Studio creates the dataset, which then becomes visible in the Data Sources pane. To create form elements from a table in a dataset Using Visual Studio you can specify the default controls to associate with each column in a database table and then simply drag and drop the table columns or even the entire table onto a form to easily create and setup data bound controls. You will be making use of these features to build a form that users can use to view and modify database values. Expand the Orders table to display the list of table columns. Select the drop-down box on Orders and choose Details, as shown in Figure Choosing Details will generate individual controls for each column in the table when the table is placed on a form. Drag the Orders table and drop it onto the form. Visual Studio creates controls for each of the columns in the table, a database navigation control, and all necessary data bindings used to link these controls to the table. This results in the form shown in Figure This causes Visual Studio to create a binding between the Customers table and the ComboBox so that the selections listed in the ComboBox drop-down box are automatically populated with the contents of the Customers table. Rather than displaying the actual customer numbers, the drop-down box will display the last name of each customer. This occurs because Visual Studio automatically modifies the data bindings so that the display value is set to the first string column in the table that is not part of the primary key. In the case of the Customers table, the first string column that is not part of the primary key is the Last Name column. Using the same technique as the previous step, create the data bindings to populate the Employee ID ComboBox drop-down box and Ship Via ComboBox drop-down box with the contents of the Employees table and the Shippers table respectively. Enlarge the form on the right side, so that there is room for you to place a PictureBox control. Drag the PictureBox control from the Toolbox to the empty area on the right side of the form. Click the smart-tags arrow button at the top right of the PictureBox to display the smart-tags menu. When you set the Size Mode property to AutoSize, the PictureBox automatically resizes based on the size of the image displayed. Visual Studio then automatically data binds the PictureBox to the Photo column. The alternate technique of first placing the PictureBox on the form and then dragging the Photo column from the Data Sources pane onto the PictureBox is included in this tutorial to demonstrate that you can first layout the controls on a form and then associate the data bindings afterwards. When the application appears, you should see the form fully populated with data and the photo of the employee associated with the order, as shown in Figure This will save your

changes to the database. Close the form to exit the application. Notice that the value for Ship Country is now Italia indicating that the changed data was properly saved to the database. You can also use the navigation controls at the top of the form to browse through database entries, add new database entries, and delete database entries. The Server Explorer pane is displayed, as shown in Figure Other data connections that you previously created might also be available. Using the Server Explorer, you can view the contents of any of the tables in the database. To view the contents of the Orders table, right-click on the Orders table and select Open. The data in the Orders table is now displayed in a table format, as shown in Figure The table view maintains an active connection to the database; therefore, you can both view and modify the data in the Orders table. By using a separate copy of the database file, you can modify the copy of the database file as much as you would like during application testing without modifying the original version of the database. To view the database file that the application is using you must create a new Server Explorer connection that references the database file in the output directory. By default, Visual Studio copies the original database file to the output directory only if the database file does not already exist in the output directory or if the original database file has been updated more recently than the copy in the output directory. If you would like Visual Studio to copy the original database file to the output directory every time you start your application, right-click on the database file in the Solution Explorer pane, choose Properties and then set the Copy To Output Directory property to Copy Always. Close the Orders table window. Still in the Add Connection dialog box, click Create. For example, to create the database file in a folder named C: However, you are encouraged to use the. Check Encrypt in order to add security to the database. In order to create an encrypted database, you must provide a password for the database. Type the word password into both the New Password and Confirm Password fields. As shown in Figure 17, a dialog box indicating that the password chosen does not meet minimum security requirements is displayed. For the purposes of this tutorial, you can leave the database password as password. The password minimum-security-requirements warning dialog box Click on the picture for a larger image Click Yes to continue using the selected password. The Add Connection dialog box should now appear, as shown in Figure Click OK to create the database connection. Visual Studio creates a connection to the new database that will now become visible in the Server Explorer pane. Adding a connection to a new password-protected database To create tables in a database In the Server Explorer pane, expand the NewDatabase. Right-click on Tables and select Create Table. The New Table window is displayed. Type Customer in the Name field to set the name of the new table. Add a column by clicking in the cell under the heading Column Name and typing the column name. Name this column Id, Now, set the remaining descriptors for the column: The Id column will be used as a primary key field. Add a column named Name and leave its associated values as the defaults, as shown in Figure Adding columns to a new table Click OK to create the table. Visual Studio creates the Customer table. The Customer table is now visible in the Server Explorer pane, as shown in Figure The newly created Customer table Right-click on the Customer table and select Open. An empty table listing the data in the Customer table is displayed. Add a record to the Customer table with as the Id and Fred as the Name. Add another record to the Customer table with as the Id and Barney as the Name. You have now successfully added two records to the Customer table, as shown in Figure The Customer table, containing new entries Close the view of the Customer table by clicking the X in the top right-hand corner of the view. The data for each row was saved to the Customer table when you moved the cursor to the next row; therefore, there is no need to explicitly save the table before closing the view. Right-click on Tables and select New Query. The Add Table dialog box is displayed listing the tables that are available in the current database. In the case of this database, the only table that is available is the Customer table. Select the Customer table and click Add to make it part of the query. The Query Designer displays the Customer table listing the table columns. Click Close to close the Add Table dialog box. Check the box to the left of the Id and Name columns in the Customer table to indicate that the query will return these columns. Notice that the Query Builder displays the query text that corresponds to the table and columns you have selected. In some cases it is easier to simply type part of the query text directly rather than always use the Query Designers graphical interface. The Query Designer automatically prompts you to enter the parameter values when you execute the query. A prompt requesting the value of the name parameter is displayed. Click

OK to see the query results. The query results are then displayed, as shown in Figure Query results being displayed Close the query designer window by clicking the X in the top right-hand corner of the window. Performing database maintenance The goal of this last section is to simply make you aware of the maintenance features that are available for SQL Server Compact Edition databases.

4: Sample Resumes - CV: C# Developer

It covers the www.enganchecubano.com technology, that is C#, www.enganchecubano.com, Web Services, www.enganchecubano.com, the major classes of the class library as well as the fundamentals of the CLR and the most www.enganchecubano.com tools. It includes also the new features introduced www.enganchecubano.com

Summary Types of N-Tier Applications There are many types of n-tier applications that programmers have developed over the years. Although all of them are clever, there has never been a consensus on how every n-tier application should be constructed. As many programmers as there are in the world, there seem to be that many methods of developing n-tier applications. Below is a list of some of the possible ways. All updates are performed in the ADO recordset, then passed back to the component for batch updating. Create a component that has many classes based around business processes. All data for the business process is passed to this component. The component will update the appropriate tables from the data supplied. Another component would be used to return views of data needed to support the user interface for a business process. Create one class per table using ADO embedded into the class. Create one class per table using ADO on the client side. Create two classes per table, one with properties that can be set from the client side EXE. The XML contains instructions on how to gather or modify the data, and a disconnected ADO recordset is returned back to the client-side class. Create one class per table using a XmlDocument object on the client side to process all the data. The client-side class bundles up properties into an XML string that is then sent to a server-side data class. The server side extracts the instructions the SQL from the XML and performs the appropriate action on the database server. The server-side class then returns XML to the client-side class to inform the client what happened on the database. In this scenario, there is only one data class for all tables. Create one server-side class for all tables. All forms then use these ADO recordsets for processing data. As you can see, there are many different methods for creating an n-tier application. They all work, and they each have advantages and disadvantages. The goal of this document is not to dispute any of these, nor to look at the advantages and disadvantages, but to simply present a way to create n-tier data classes within Visual Basic. It also provides you with properties that match up to each column in the base table. You use a standard data adapter object to fill up the DataSet from the data source. The wizard that generates this Typed DataSet reads the schema information from the data source and maps these data types to each of the columns. This is why this is called a Typed DataSet. A Typed DataSet will help you speed up your development process in a few ways. This avoids run-time errors as column names can be checked at compile time. Second, you no longer have to see SQL in your front-end client application. All of the SQL is buried in the data adapter. By putting these Typed DataSets into a separate component, you are able to reuse these classes from multiple projects. There are many goals that an n-tier application design should achieve. Here are some of them. If you change the underlying data access methods, the client-side code should not have to change. All data access routines should be exposed as objects instead of function calls. SQL should be eliminated from the client-side code. The client code should just be concerned with methods and properties. Table and column names should be eliminated from the client-side code. Typed datasets can present table and column names as properties, providing an IntelliSense list, as opposed to having to type in a string name. This means at compile time, checks can be made for data types and names of columns. The client code should not care where the data comes from. It should just care that it can retrieve and modify the data in some object and the object will take care of the details. The coding you need to do on the client side should be simplified. Instead of using many functions, your application should be able to use objects with properties and methods. It becomes easier to create and use the classes than the function calls. It becomes easier to add functionality to your applications, and change the functionality, without breaking the client-side code. Disadvantages to N-Tier Although there are many advantages to a good n-tier application, there are some disadvantages as well. You end up creating a lot of classes. This can lead to maintenance issues and could even be a performance issue as it does take time to create a new class at run time. N-tier does not work well when you do not know the structure of the tables from which you will be retrieving data. For example, in a Query By Example QBE application where the user

may put together several columns from several tables, there is no way to generate classes on the fly to accomplish this. Creating reports is not something that lends itself to a good n-tier design, as report writers do not use classes to get at data. In the end, the advantages of a good n-tier design will far outweigh the disadvantages. There is certainly nothing wrong with mixing both of these paradigms in the same application if appropriate.

Creating an N-Tier Application

When you talk about a true distributed n-tier type of application, you are talking about separating the components of the different tiers on different machines as well as in separate components. Figure 1 shows a typical example of an n-tier application with multiple components on each machine. A distributed n-tier application has three physical tiers with one or more logical tiers on each machine. There are many different ways you could configure an n-tier application. For example, the business rules may go on a separate machine and you might use .NET Remoting to talk from the client application to the business rule tier as shown in Figure 2. Business rules can be placed on a separate machine to facilitate ease of maintenance. You may also have a data input validation rule component on the client to check simple rules such as required fields and formatting. These are rules that you do not want to make a trip across the network just to check. You may then also add a business rule layer on the same tier as the data layer component to check complicated business rules that compare the data from one table to another. These are just a few different configurations that you may utilize. Of course, you could come up with something unique that fits your specific situation. Regardless of how you structure the physical implementation of the components, make sure that the logical structure of the program is broken up into components as shown in the above figures.

Creating the User Interface

In the example you see in Figure 1, the client tier consists of a Windows application and a business rule component. The Windows application makes all requests for data, and all updates through the business rule component. This isolates the location of the data from the Windows application. The advantage of doing this is if you change where the data comes from, you do not need to make any changes to the client application, only to the business rule component. This is a DataGrid that has been bound to the return result from the business rules component.

Perform the following steps to build a simple Windows client application that will display employee information in a DataGrid control on a Windows Form.

- Create a new Windows Application project named EmpClient.
- Rename the default form Form1.
- Drag a DataGrid onto this form. Set the Name property to grdEmps.
- Add a Button control to this form. Set the Name property to btnUpdate.
- Set the Text property to Update.

At this point, the user interface for your employee form is complete. Now it is time to start building the components so you can retrieve the data to populate this DataGrid.

Creating the Data Tier

The data tier is responsible for connecting to your data source, building a typed data set, and returning that data set from a method within this component. Follow these steps to build a data tier component.

- In the Solution Explorer window, right-click the solution named EmpClient.
- On the shortcut menu, click Add, and then click New Project.
- Choose the Class Library template.
- Set the name of this class library to EmpData.
- Delete the class file named Class1.
- To add a new component to the project, on the Project menu, click Add Component.
- Set the name of the component to clsEmp.
- View the code for this component and change the name of the class from clsEmp to Employees.
- In the design view, click and drag a SqlDataAdapter control from the Data tab of the toolbox onto the design surface of this component.
- Go through the steps of this wizard to connect to your SQL Server, pointing to the Northwind database on that server.
- Select all rows and columns from the Employee table within this database.
- Set the name of the New DataSet to dsEmps.

At this point, you have a component with some data access objects on it. The reason to use a component instead of a regular class is that you need the ability to drag a Connection and DataAdapter object onto a design surface. A component will let you do this; a regular class will not. Of course, you could always just create your own data adapter and connection objects in code, but this way is much easier. All that is left to do after adding this component is to add a couple of methods to the component.

5: Web Application Development using Visual C++

This is a sample telephony application which will help you to develop applications using TAPI API and C#.NET. Here basic knowledge about TAPI is assumed.

Microsoft and the Web. Windows on the Desktop. A New Programming Platform. The Role of XML. Problems of Windows Development Applications of the Future. NET and Console Applications. Hello World in VB. Performing Calculations in VB. Arrays and Parameterized Properties. Inheritance and Exceptions in VB. Review of Object-Oriented Concepts. Acme Travel Agency Case Study: Windows Applications Using the. Windows Forms Event Handling. Under the Hood of a VS. Setup and Deployment Projects. Input and Output in. Garbage Collection and Finalization. Device-Independent Graphics and Abstraction. Parameters and Stored Procedures. Database Transactions and Updates. Acme Travel Agency Case Study. NET and Web Forms. Web Applications Using Visual Studio. Database Access in ASP. WebService Class and Visual Studio. Platform Invocation Services Plnvoke. Overview of Visual Studio. Creating a Console Application. Enabling Debug and Trace Output. Using the Debug and Trace Output. Using the Debug and Trace Classes. Using Switches to Enable Diagnostics. Enabling or Disabling Switches. Preface Preface For many years, Microsoft Visual Basic has been used as the ultimate rapid application development tool for Windows applications. Its ease of use revolutionized Windows programming, and successive generations of Visual Basic have progressively made it more powerful. NET brings a sea change to software development. The powerful Common Language Runtime and the vast. NET Framework class library provide a consistent software platform for all. NET languages almost seamless, while the power of each is such that for most projects you will probably never have to use a mixed language approach, unless other factors such as legacy code or programmer skill sets steer you in that direction. There is substantial change to the Visual Basic language itself, and VB6 code will not run unmodified in the. Also, the new version of the language, Visual Basic. NET, or just VB. NET, is now a fully object-oriented language with features such as interfaces, inheritance, and polymorphism. The result is that there is a definite learning curve when moving to VB. And learning the new programming language is only part of the challenge. The much greater challenge is learning the. This book is written for the experienced programmer to help you quickly come up to speed on the VB. NET language and then go on to an in-depth study of the. It is a practical book for practicing professionals, and it has many examples and a realistic case study that continues through many of the chapters. The goal is to equip you to begin building significant applications using Visual Basic. The book is part of The Integrated. Other books in the series provide a more a more basic introduction to VB. NET, discuss the issues of migrating to VB. NET, and cover other important. NET languages and topics in the Framework. See the front of this book for a list of titles in the series. This book, in substance and structure, is quite close to the companion titles Application Development Using C and. A major difference between those books and this, besides using VB. Organization The book is organized into six major parts and is structured to make it easy for you to navigate to what you need to learn. Part 1, consisting of the first two chapters, provides an overview that should be read by everyone. It answers the big question, What is Microsoft. Part 2, consisting of Chapters 3 to 6, covers the VB. Even if you know classic Visual Basic, you should read this part, paying attention to the changes in data types Chapter 3 and the new object-oriented features Chapters 5. Chapter 6 covers important interactions between VB. The Acme Travel Agency case study, which is elaborated throughout the entire book, is introduced in Chapter 5. Part 3, consisting of Chapters 7 and 8, covers the fundamentals of Windows Forms. Windows Forms is a set of classes in the. NET Framework for writing graphical user interfaces. Programmers familiar with previous versions of Visual Basic will notice that this significantly changes the programming model, yet also introduces flexibility not previously available. Part 4, consisting of Chapters 9 and 10, introduces important fundamental topics in the. Chapter 9 discusses assemblies and deployment, which constitute a major advance in the simplicity and robustness of deploying Windows applications, ending the notorious situation known as "DLL hell. NET Framework classes and covers the topics of metadata, serialization, threading, attributes, asynchronous programming, remoting, and memory management. Part 5,

consisting of Chapters 11 and 12, covers additional advanced topics in UI programming with VB. Chapter 12 introduces some important additional advanced topics, such as visual inheritance, MDI, and the use of ActiveX controls in. Part 6, consisting of Chapters 13 to 17, covers important parts of the .NET Framework that are useful in creating a variety of different applications. Chapter 13 covers ADO. Chapter 14 introduces the fundamentals of ASP. Chapter 15 covers SOAP and Web services, which provide an easy-to-use and robust mechanism for heterogeneous systems to interoperate. Chapter 16 covers the topic of security in detail, including code access security and declarative security. Chapter 17 covers interoperability of. Appendices introduce Visual Studio. NET and the debug and trace classes provided by. Sample Programs The only way to really learn a major framework is to read and write many programs, including some of reasonable size. This book provides many small programs that illustrate pertinent features of .NET in isolation, which makes them easy to understand. The programs are clearly labeled in the text, and they can all be found in the software distribution that accompanies this book. A major case study, the Acme Travel Agency, is progressively developed in a number of the chapters, beginning with Chapter 5. It illustrates many VB. NET Framework features working in combination, as they would in a practical application. When expanded, a directory structure is created, whose default root is c: The sample programs, which begin with the second chapter, are in directories Chap02, Chap03, and so on. All the samples for a given chapter are in individual folders within the chapter directories. The names of the folders are clearly identified in the text. An icon in the margin alerts you to a code example. Each chapter that contains a step of the case study has a folder called CaseStudy, containing that step. If necessary, there is a readme. As part of The Integrated.

6: How to Create a C# Windows Forms Application

For example, Cocoa, using Objective C licensed by Apple, Ruby On Rails, using Ruby licensed by MIT and Grails using JAVA licensed by APACHE. We could continue with so much more, but the one we are focusing on in this tutorial is Microsoft www.enganchecubano.com MVC, using C#.

Application Development with C and .NET. October 3, This course covers the whole Microsoft .NET. It is an advanced course for students who are already familiar with object-oriented programming and basic Web programming techniques. It comes with a full set of Powerpoint slides both in English and in German and with an extensive case study. Text books with exercises and sample solutions are also available. All material is published under the Microsoft Curriculum Licence. .NET in the near future they feel the need to provide their students with a profound overview of the .NET technology, which becomes more and more important in industry. A course with this goal is different from introductory programming courses, since it can assume that the students are already proficient in programming and know the basics of Web engineering etc. Our course was designed with these goals in mind. It covers the whole .NET, the major classes of the class library as well as the fundamentals of the CLR and the most important. It includes also the new features introduced in .NET 2.0. The course has been delivered several times at the university of Linz. Parts of the course have also been given as industry seminars and tutorials. There are almost 100 slides, more than exercises with sample solutions and a case study involving C#, ASP.NET and Web Services. Learning Objectives The course will teach students the following skills: Programming in C# using high-level abstractions such as classes, interfaces, namespaces, properties, delegates, events, threads, attributes, generics, and iterators, new features of C# 3.0. It also covers the most important features of the class library collections, IO, threading, sockets, Windows Forms, reflection, Xml, generics, Ability to compare the .NET platform with the Java technology. Contents The course can either be delivered as a whole or as two separate courses on C# and .NET. In total, it has 30 lecturing units of 45 minutes each corresponding to 2 units per week or 3 ECTS credit points during a semester. We recommend to accompany the course with a programming lab of at least 15 units, which can be organized either as weekly exercises or as a term project. Exercises can be found in our text books. Sample solutions are on our web site. .NET The Programming Language C# Types, expressions, declarations, statements, classes, structs, inheritance, interfaces, delegates, exceptions, namespaces, assemblies, attributes, native calls, threads, XML comments, genericity, anonymous methods, iterators, partial types. .NET Dynamic web pages, Web Forms, event handling, Web Controls, validators, user controls, custom controls, state management, configuration, web page design with Visual Studio. .NET Framework Tools Assembly cache viewer, native image generator, GAC utility, strong name tool, code access security policy tool, web services description language tool We have written two text books on C#. Both are available in English and in German. Our slides have been tailored to these books.

7: Develop for PCs – Windows app development

Application Development using www.enganchecubano.com in C# or VB. Create the application and/or a web site for a business (choice is yours). You must use three (3) programming techniques discussed in this class.

Net Framework C Windows Forms C programmers have made extensive use of forms to build user interfaces. Each time you create a Windows application, Visual Studio will display a default blank form, onto which you can drag the controls onto your applications main form and adjust their size and position. The first step is to start a new project and build a form. Enter a project name at the bottom of the dialouge box and click OK button. The following picture shows how to create a new Form in Visual Studio. The Windows Form you see in Designer view is a visual representation of the window that will open when your application is opened. You can switch between this view and Code view at any time by right-clicking the design surface or code window and then clicking View Code or View Designer. The following picture shows how is the default Form Form1 looks like. At the top of the form there is a title bar which displays the forms title. Form1 is the default name, and you can change the name to your convenience. The title bar also includes the control box, which holds the minimize, maximize, and close buttons. If you want to set any properties of the Form, you can use Visual Studio Property window to change it. If you do not see the Properties window, on the View menu, click Properties window. This window lists the properties of the currently selected Windows Form or control, and its here that you can change the existing values. For example , to change the forms title from Form1 to MyForm, click on Form1 and move to the right side down Properties window, set Text property to MyForm. Then you can see the Title of the form is changed. Likewise you can set any properties of Form through Properties window. You can also set the properties of the Form1 through coding. For coding, you should right-click the design surface or code window and then clicking View Code. How to Pass Data Between Forms In C , there are many situations the new programmers face the same problem about how to pass data and values from one form to another. The following link will guide you This is in contrast to single document interface SDI applications, which can manipulate only one document at a time. Copy and paste the following C source code to source code editor of your Visual Studio.

8: C# Corner - A Social Community of Developers and Programmers

It is very disappointed for Visual C++ programmers having years of experiences in C++, who can not do web application programming in Visual C++ like C# programmer do using www.enganchecubano.com And very difficult for C++ programmers to shift from C++ to C#/www.enganchecubano.com for web application development.

Hence it can also be useful for VB. Here basic knowledge about TAPI and other related terms is assumed. To have more knowledge on TAPI 3. Background Hello, myself Gohel Devang M. This is my first attempt to put some sample code on any site so if you do have any problems mail me at devang. This is a sample code to interface TAPI 3. NET platform and C as language. So people interested in developing telephony applications using C will find this very useful. I was inspired to do this because I was not able to find such code on this site. To do that first create a new project or open an existing project, then right click on the solution file in Solution Explorer. This will open a dialog box showing three tabs as below: Then click on Browse button and select the tapi3. Then press OK in the dialog box that had popped when you select Add references from the right click popup menu of the Solution Explorer. Now you are ready to work with TAPI 3. The code below is a declaration of the TAPI object and addresses the interfaces that will hold the addresses which are responsible for call handling, and basic call control interface which will hold the reference to the object that will be responsible for handling basic operations of the call. The main functions are: Initialize will initialize TAPI. EnumerateAddresses will give the list of available TSPs. For that you need to select the line on which you want to receive calls and press the Register button. This is specially designed according to the requirements of the application. To do IP calls or H. Otherwise it will not succeed in calling to the remote destination. How to answer an incoming call The incoming calls will give notification in the call status area. Then according to whether you want to accept or reject the call, you check the checkbox named Reject to reject incoming calls, and press Answer or simply press Disconnect. To accept calls, do not check Reject checkbox and simply press Answer button which will connect to the call. How to transfer a call To transfer a call, first there should be one active call existing. Then you can specify the address to which the call is to be transferred to, as shown in the figure: Here I have specified the internet address since the call was an IP call. Refer the MSDN documents for more information on that! NET as the platform. I am also interested in J2EE application development. My other areas of interest are: History Latest revised version. This is the first release of this code so if you do have any suggestions they are always welcomed by me. License This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below. A list of licenses authors might use can be found here Share.

9: Rapid Application Development Using SQL Server Compact Edition and Visual C#.NET

MonoDroid is a software product developed by Novell to build Android based mobile applications using C# www.enganchecubano.com In this tutorial, we will learn how to build our very first Android based application using MonoDroid and Visual Studio

In the following examples, we will show how to create an MFC app, so this optional component was installed. Out of the box, Visual Studio can open any folder of code and be configured to build using CMake, a cross-platform build system. MSBuild is a robust and fully featured build system that allows building projects in Visual Studio that target Windows. Building an MSBuild-based project just requires a. In Visual Studio, you can also simply open a folder of code files and immediately begin working in it. In the background, Visual Studio will index your files and providing Intellisense support along with refactoring and all the other navigation aids that you expect. You can create custom. Creating new projects If you are creating a new project from scratch, then you can start with one of a variety of project templates.: Each template provides customizable build configurations and boilerplate code that compiles and runs out of the box: Project templates are included for each of these types of desktop applications depending on the features you select for the workload. Project Wizard Once you have selected a template, you have the option to customize the project you have selected to create. Each of these project types has a wizard to help you create and customize your new project. The illustrations below show the wizard for an MFC application. The wizard creates and opens a new project for you and your project files will show up in Solution Explorer. At this point, even before you write a single line of code, you can build and run the application by pressing F5. Editing code and navigating Visual Studio provides many features that help you to code correctly and more efficiently. Whether it be the powerful predictive capabilities provided by IntelliSense or the fluid navigation found in Navigate To there is a feature to make almost any action faster inside Visual Studio. Let Visual Studio do the work for you with autocompletion simply by pressing Tab on the item you want to add from the member list. You can also hover over any variable, function, or other code symbol and get information about that symbol using the quick info feature. Peek Definition allows you to view the code that defines the selected variable without even having to open another file which minimizes context switching. We also have support for some of the more common refactoring techniques like rename and extract function that allow you to keep your code looking nice and consistent. Debugging and Diagnostics Debugging applications is what Visual Studio is famous for! With a world-class debugging experience that provides a plethora of tools for any type of app, no tool is better suited to debugging applications that target the Windows desktop platform. View data values from your code with debugger data tips. Take memory snapshots and diff them to reveal potential memory leaks, and even invoke PageHeap on your app from inside Visual Studio to help solve the notoriously hard problem of memory corruption. Track live CPU and memory usage while your application runs and monitor performance in real-time. Testing your code Unit testing is a very popular way of improving code quality, and test-driven-development is fully supported inside Visual Studio. Writing unit tests is easy and can help find problems as they arise instead of later on when things are harder to isolate. Working with others Besides all of the individual developer activities that Visual Studio makes more productive, collaboration is also something that is directly integrated into the IDE. Visual Studio Team Services is a suite of features that optimize the team collaboration process for software development organizations. Create work items, track progress, and manage your bug and open issue database all from inside Visual Studio. Git is fully supported and works seamlessly with the Team Explorer, allowing for easy management of branches, commits, and pull requests. Simply sign up for a Visual Studio Team Services account for free, then you can track the source code of your desktop applications into Visual Studio Team Services. Visual Studio Team Services also simplifies continuous integrations for your desktop applications. Create and manage build processes that automatically compile and test your apps in the cloud. Windows Store packaging for desktop apps When you are ready to deploy your desktop application, you would typically build an executable. This allows you to easily distribute your application however you like, for example via a download from your website or even through a

third-party sales platform such as Steam. A new option for Windows desktop apps is to be available in the Windows Store with all the advantages that entails. When targeting Windows 10, this can provide advantages including streamlined deployment, greater reach, simpler monetization, simplified setup authoring, and differential updates. Download Visual Studio , try it out and share your feedback. For problems, let us know via the Report a Problem option in the upper right corner of the VS title bar. Track your feedback on the developer community portal. For suggestions, let us know through UserVoice.

Peasants, ideology, and new incentive systems, Jiangsu Province, 1978-1981 David Zweig Westward with the Sun Maternal-Newborn Child Nursing Hitlers naval war Family government. Edinburgh Characters Pt. 1. ACOA Executive Committee minutes and National Office memoranda, 1952-1975 (6 reels) Maersk annual report 2016 Entry restrictions and Japanese lawyers incomes in international legal practice (Report Rand Corporation) Issues in U.S.European Union Trade Morphys games of chess Terry Eagleton Sandra M. Gilbert and Susan Gubar Penny Boumelha Sally Shuttleworth Susan Meyer Sharon Mar Understanding food science and technology Eric foner give me liberty ap edition The Unfinished Struggle Opening and Closing a PowerPoint Presentation Codes, military culture, and clubmen in the English Civil War Marie Antoinette (MTI) Indianhead ridge and beyond Shadow Dance (An Avon Romantic Treasure) Interlocking pieces Molly Gloss Eugene Kranz returns Apollo 13 to earth Blind Josef Pronek dead souls Catalogo de productos total life changes Gunner Asch goes to war Appendix 2 : Legalism in the book of Galatians Cardiology and neurology Sobriety checkpoints and blanket patrols reduce alcohol-related crashes National Hardcore Drunk Driver Pr Goethe and the novel India cabinet ministers list 2017 in gujarati language Jeet Kune Do, a New Era Book of sith secrets from the dark side Death of a Hallow Man (Chivers Sound Library) Upside down, inside out and backwards, or, Downside up, outside in and frontwards Buddhist views of nature Robert A.F. Thurman Australian Dictionary of Biography, Volume 16 Sedra and smith 7th edition solutions Avoidance of love: a reading of King Lear. Pennys guide to teenage charm and popularity Sounds of Silver Doves Cry