C++ Builder 2010 Professional Getting started.

Kjell Gunnar Bleivik: http://www.kjellbleivik.com/ October 18. 2010. Fixed broken link.

Status: Most probably finished. Look at the date in case more should be added.

Follow me on Twitter and join my C++ Builder group on Facebook:

Twitter: http://twitter.com/kbleivik

FaceBook: http://www.facebook.com/people/Kjell-Gunnar-Bleivik/1244860831

Mini network: http://www.digitalpunkt.no/

## 1. Installing C++Builder 2010, the help system etc.

I ordered my upgrade of Borland C++ Builder Prosessional 2010 on September 21 2009 so I could choose an additional free product. I choose Delphi PHP 2.0, since I am fairly used to PHP. In order to install C++ Builder 2010, I had to upgrade my 2009 version. I have made an 2009 and 2010 upgrade shortcut on my desktop. You should find your upgrade program:
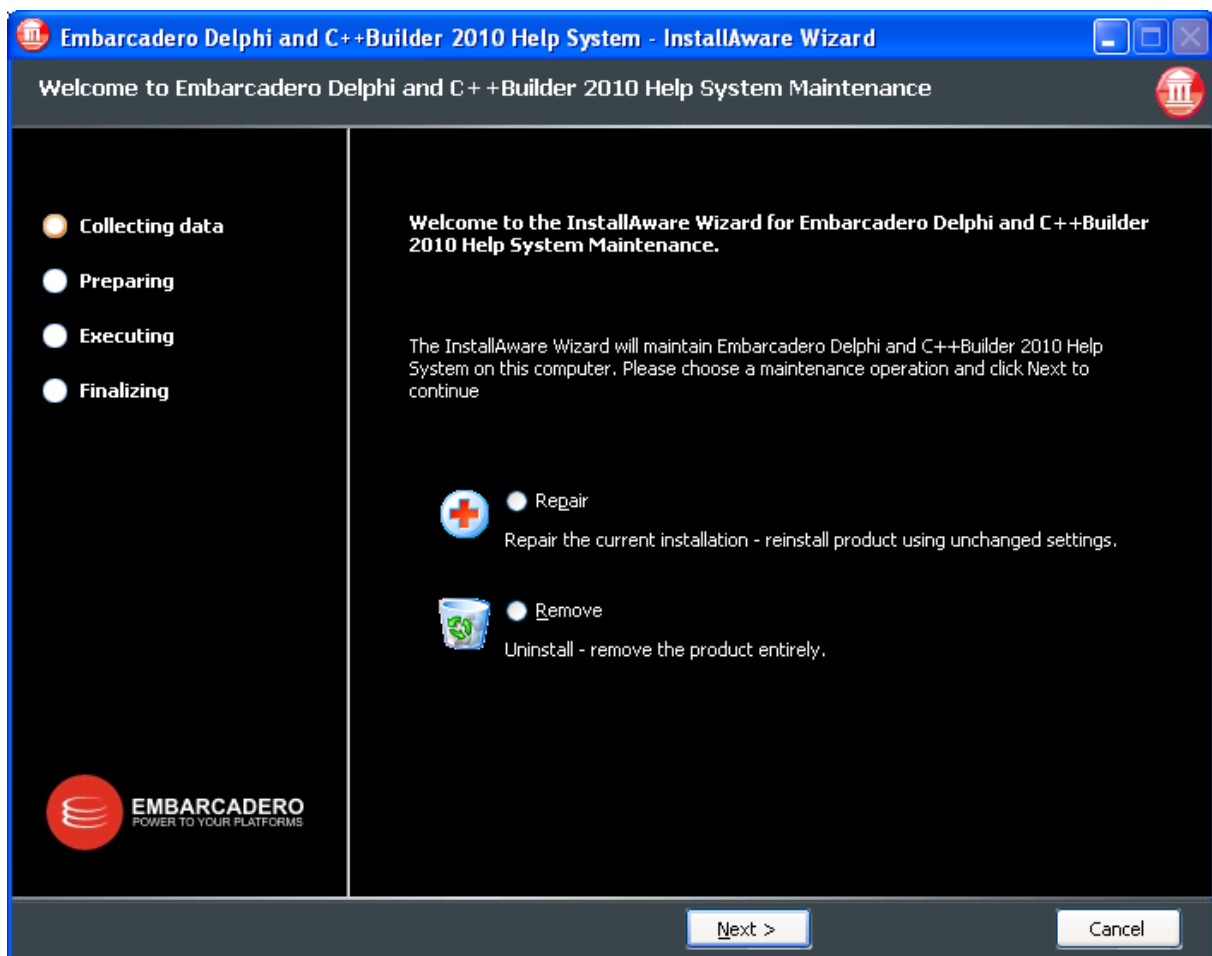
| your start menu or in | the all program category | CodeGear RAD studio 2009 | Check for updates | Program |

When finished upgrading the 2009 Builder, I could run the C++ Builder 2010 Setup program. In addition, I installed the additional first three programs that I also find in the Install Folder. Look at the screendumps below, so you get it correct.

- Help_Setup Program
- dbpack_setup Program
- boost_setup Program
- Additional_Products HTML document.
- ERStudio_Interbase HTML document
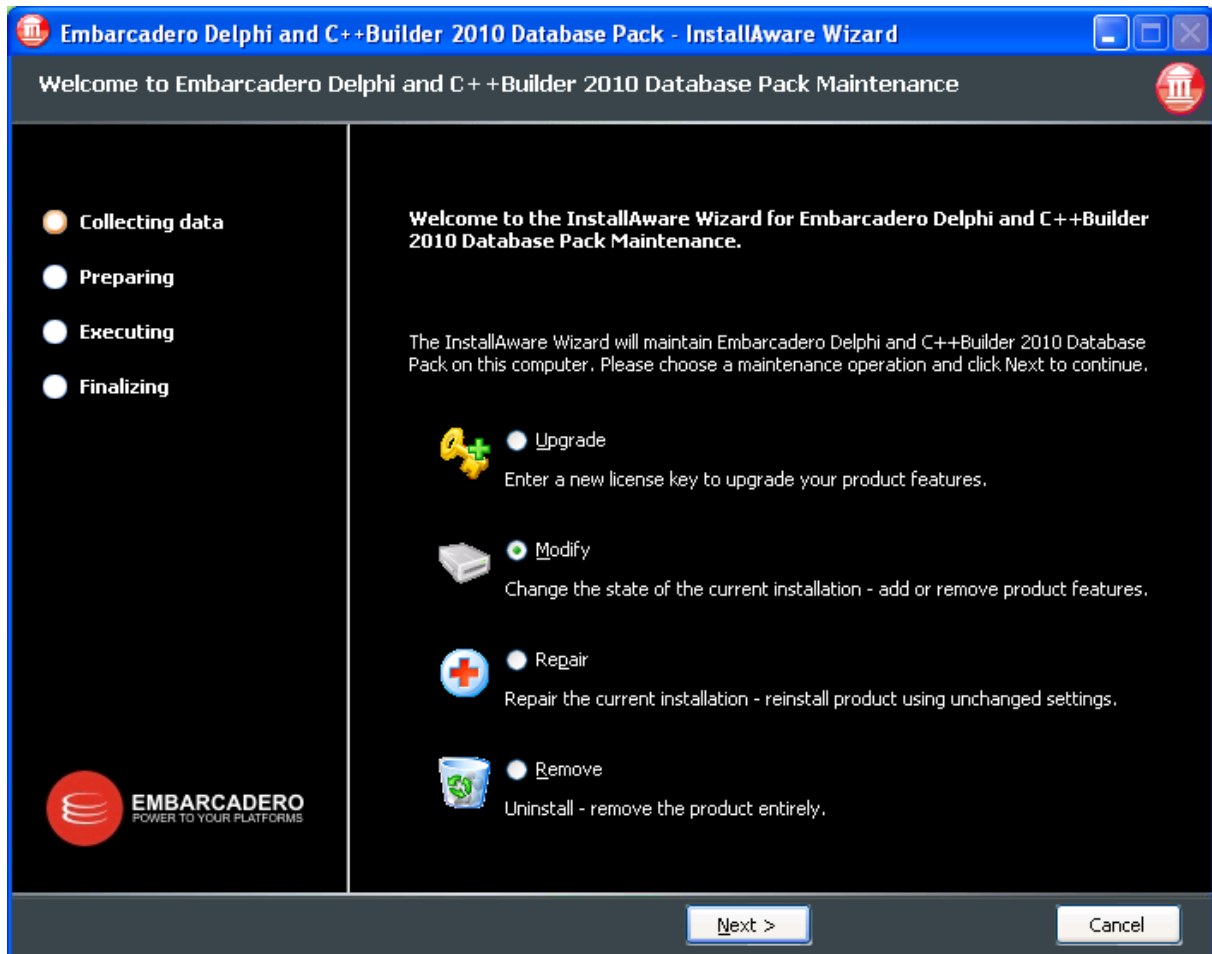
2. Getting started with C++ Builder 2010 Professional.

 If you learn to use the welcome page efficiently, that may be all you need.   On the "documentation" menu, you should start with, yes "Getting started" and then "RAD Studio Help" and so on. As an example click: | Documentation | …
and try to locate this http://docs.embarcadero.com/products/rad_studio/ page with Wiki pages, PDF documents, zipped code examples for download, PDF documents since C++Builder 6 (scroll down to the bottom) and CHM http://en.wikipedia.org/wiki/Microsoft_Compiled_HTML_Help files. There has been questions on the Embarcadero forum "What's New in the Help, and Where is Everything": ms-help://embarcadero.rs2010/rad/What's_New_in_the_Help,_and_Where_is_Everything.html
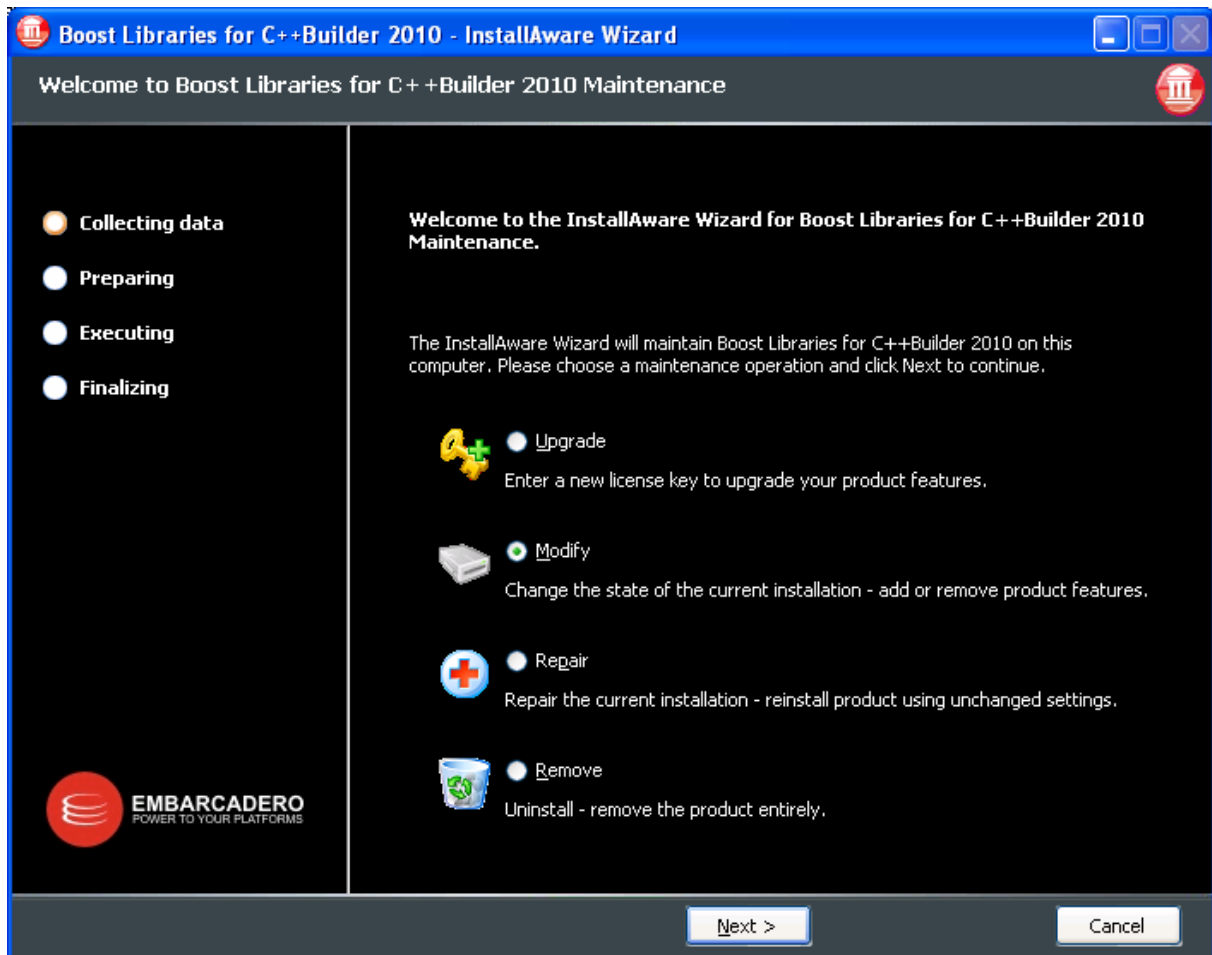
I have written this document that assumes that you have already read, this http://www.oopschool.com/books/CPB2009.pdf document in the hope of giving you a flying start so you don't drown here: http://docwiki.embarcadero.com/CodeSamples/en/Category:C%2B%2B or on similar pages.

3. Writing and compiling traditional windows API programs with C++ Builder.

You find the explanation in this

http://www.webproworld.com/programming/90952-walk-garden-part-3-some-useful-c-hints.html#post464778

WPW thread. In case that tread dies, I repeat the procedure and the code here:

The C++Builder is a visual programming platform. In a sense you draw your program('s API) by dropping binary components on a form and modify them through the object inspector. The non visual part is writing your programs event handlers. But if you want to make traditional Windows API applications, that is also very easy. You do it like this:

1. | File | New | VCL Forms Application - C++ Builder |
2. | Project | Remove from project | Unit1.cpp Form 1 | OK | (Don.t forget to click Unit1.cpp Form 1 before hitting OK)
3. | View | Units... | OK | (Choose the default name the first time you try).
4. Delete all the code in Project1 and replace it with the following example code:

```cpp
#include <vcl.h>  // Note this line may be forgotten in example code and
you get a linker error.
#define STRICT
#include <windows.h>
#pragma hdrstop

LRESULT FAR PASCAL _export WndProc(HWND, UINT, WPARAM, LPARAM);


int PASCAL WinMain(HINSTANCE hInstance,
  HINSTANCE hPrevInstance, LPSTR lpszCmd, int nCmdShow)
{
  static char AppName[] = "HelloWorld";
  HWND        hwnd;
  MSG         msg;
  WNDCLASS    wndclass;
  if (!hPrevInstance)
  {
    wndclass.style         = CS_HREDRAW | CS_VREDRAW;
    wndclass.lpfnWndProc   = (WNDPROC)WndProc;
    wndclass.cbClsExtra    = 0;
    wndclass.cbWndExtra    = 0;
    wndclass.hInstance     = hInstance;
    wndclass.hIcon         = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hCursor       = LoadCursor(NULL, IDC_ARROW);
    wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
    wndclass.lpszMenuName  = 0;
    wndclass.lpszClassName = AppName;

    RegisterClass(&wndclass);
  }

  hwnd = CreateWindow(AppName,
    "Hello World",
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    NULL,
    NULL,
    hInstance,
    NULL);

  ShowWindow(hwnd, SW_NORMAL);

  while (GetMessage(&msg, NULL, 0, 0))
  {
    TranslateMessage(&msg);
    DispatchMessage(&msg);
  }
  return msg.wParam;
}

LRESULT FAR PASCAL _export
WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
  switch(message)
  {
    case WM_PAINT :
    {
      char text[] = "Hello World!!";
```

```
    PAINTSTRUCT ps;
    BeginPaint(hwnd, &ps);
    TextOut(ps.hdc, 20, 20, text, 13);
    EndPaint(hwnd, &ps);
    break;
  }
  case WM_DESTROY : {
    PostQuitMessage(0);
    return 0;
  }
  default:
    return DefWindowProc(hwnd, message, wParam, lParam);
  }
  return 0;
}
```

Related search http://www.amazon.com/ : Charles Petzold Programming Windows

Exercise 3.1:

http://rkaiser.de/rkaiser/CB2007_Buch/CB2007_Buch.htm

Solve problem 10.13 page 1079 in that book that should be found in the

http://rkaiser.de/rkaiser/CB2007_Buch/CB2007_AufgLsg.pdf (with 589 pages)

Aufgaben und Lösungen PDF document.

That is an excellent book and you do not need to read German to understand (some of) the code.

4. 3D graphics and games.

OpenGL http://www.opengl.org/

is the industry standard for high performance graphics. It can be used with the C++ Builder.

1. Read: http://www.opengl.org/wiki/Getting_started
2. Search for OpenGL in the help system with no filter. Read or scan subjects that may interest you.
3. Use Google suggest and start with the following query: opengl

   and read some subjects that interests you. As explained in the C++ Builder 2009 introductory PDF there is a tradeoff between reading and coding. Some young persons don't read at all. My experience is however that you shall not be too fast.
4. Refine the above suggest query to: opengl C++

   scan and read and finally to:
5. opengl "C++Builder" OR "C++ Builder"

   scan and read, at least the following document
   http://www.fdi.ucm.es/profesor/segundo/TOpenGL%20BCB5/OpenGLPannel.pdf

Exercise 4.1: Solve problem 10.19 page 1135 in Kaiser's book. If you have the book, you may read chapter 10.19 "3D-Grafik mit OpenGL" before you start or look at the solution on the CD.

Exercise 4.2: Here http://www.kjellbleivik.com/Books/GameDevelopment.php

you find some additional unstructured (sorry time is a limiting factor and I am 62 years old) resources. Look up the explanation of some of the C++ related books on the internet or on Amazon. One of them explains how to combine C++ with the Python scripting language to produce highly interactive games. C++ is used to produce the game engine and Python to make the program's API flexible. The Boost C++ Library: http://www.boost.org/ is distributed with C++ Builder 2010.

Note:

"C++Builder 2010 is the second release to ship with portions of the boost project. We are only shipping a subset of these libraries".

http://blogs.embarcadero.com/ddean/2009/09/23/34847

Install that library now as explained in the introduction above. Now it is time to make some games of your own. If you don't know where to start, start easy with the first book in the Python section of my above page. You may also look up some code on the internet that may help you. When you master C++, the Boost Library and Python, it is time to make your own games combining (your own) real video clips with C++ to produce advanced 3D games. Why is it important to learn C++ and Python:
http://www.webproworld.com/programming/90333-game-programming-why-c-python.html#post462438

When you make games, it is important to have (be) a good designer and a good programmer. If you don't have both abilities try to join forces with one that have the one you miss. Note that imagination is sometimes more important than knowledge, especially when you design games. But it may important to know the concept of multiple inheritance in C++ when you make games. You inherit capabilities from both your mother and father. Both insects and birds inherit properties from species that can fly. So remember to program to an interface and not to an implementation, that is start very generally and extend functionality by inheritance. A football, an orange and a planet should both inherit from and abstract Sphere base class. Last but not least, listen to your own night dreams, they may be a good source when you design a new game.

Exercise 4.3.

Read this

http://www.webproworld.com/other-engines-directories/91374-new-firefox-3d.html#post466619

thread and acquire as much information as possible about WebGL:

http://sourceforge.net/projects/webgl/

Go back to Exercise 3.1. in http://www.oopschool.com/books/CPB2009.pdf

Here is some source code (The "Source button" do not function):

I. WebBrows.cpp

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
#include <tchar.h>
//---------------------------------------------------------------------------
USEFORM("WebBrwsU.cpp", WebMain);
//---------------------------------------------------------------------------
WINAPI _tWinMain(HINSTANCE, HINSTANCE, LPTSTR, int)
{
        try
        {
                Application->Initialize();
                Application->MainFormOnTaskBar = true;
                Application->CreateForm(__classid(TWebMain), &WebMain);
                Application->Run();
        }
        catch (Exception &exception)
        {
                Application->ShowException(&exception);
        }
        catch (...)
        {
                try
                {
                        throw Exception("");
                }
                catch (Exception &exception)
                {
                        Application->ShowException(&exception);
                }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

II. WebBrwsU.cpp
```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "WebBrwsU.h"
//---------------------------------------------------------------------------
```

```cpp
#pragma package(smart_init)
#pragma link "SHDocVw_OCX"
#pragma resource "*.dfm"
TWebMain *WebMain;
//---------------------------------------------------------------------------
__fastcall TWebMain::TWebMain(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TWebMain::URLComboBoxClick(TObject *Sender)
{
        if (URLComboBox->Text != "")
                HTML->Navigate(URLComboBox->Text);
}
//---------------------------------------------------------------------------
void __fastcall TWebMain::URLComboBoxKeyPress(TObject *Sender, wchar_t &Key)
{
        if (Key == VK_RETURN) {
                Key=0;
                if (URLComboBox->Text == "") return;
                URLComboBoxClick(Sender);
        }
}
//---------------------------------------------------------------------------


void __fastcall TWebMain::URLComboBoxDblClick(TObject *Sender)
{
        if (URLComboBox->Text != "")
                HTML->Navigate(URLComboBox->Text);
}
//---------------------------------------------------------------------------
 void __fastcall TWebMain::GoBtnClick(TObject *Sender)
{
  URLComboBoxClick(0);
}
//---------------------------------------------------------------------------
void __fastcall TWebMain::StopBtnClick(TObject *Sender)
{
        HTML->Stop();
        StatusBar->SimpleText = "Done";
}
//---------------------------------------------------------------------------


void __fastcall TWebMain::ReloadBtnClick(TObject *Sender)
{
                URLComboBoxClick(0);
}
//---------------------------------------------------------------------------
 void __fastcall TWebMain::SourceBtnClick(TObject *Sender)
```

```
{
        if (HTML->Document)
                SourceBtn->Caption = "View Document";
        else
                SourceBtn->Caption = "View Source";


}
//---------------------------------------------------------------------------
```

III. WebBrwsU.h

```
//---------------------------------------------------------------------------

#ifndef WebBrwsUH
#define WebBrwsUH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include "SHDocVw_OCX.h"
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <OleCtrls.hpp>
#include <SHDocVw.hpp>
//---------------------------------------------------------------------------
class TWebMain : public TForm
{
__published:   // IDE-managed Components
        TPanel *Panel1;
        TComboBox *URLComboBox;
        TStatusBar *StatusBar;
        TWebBrowser *HTML;
        TButton *GoBtn;
        TButton *StopBtn;
        TButton *ReloadBtn;
        TButton *SourceBtn;
        void __fastcall URLComboBoxClick(TObject *Sender);
        void __fastcall URLComboBoxKeyPress(TObject *Sender, wchar_t &Key);
        void __fastcall URLComboBoxDblClick(TObject *Sender);
        void __fastcall StopBtnClick(TObject *Sender);
        void __fastcall ReloadBtnClick(TObject *Sender);
        void __fastcall SourceBtnClick(TObject *Sender);
        void __fastcall GoBtnClick(TObject *Sender);

private:        // User declarations
public:         // User declarations
        __fastcall TWebMain(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TWebMain *WebMain;
```

```
//--------------------------------------------------------------------
#endif
```

IV. WebBrws.dfm

```
object WebMain: TWebMain
 Left = 0
 Top = 0
 Caption = ' Browse Toolbar'
 ClientHeight = 236
 ClientWidth = 383
 Color = clBtnFace
 Font.Charset = DEFAULT_CHARSET
 Font.Color = clWindowText
 Font.Height = -11
 Font.Name = 'Tahoma'
 Font.Style = []
 OldCreateOrder = False
 PixelsPerInch = 96
 TextHeight = 13
 object Panel1: TPanel
  Left = 0
  Top = 0
  Width = 383
  Height = 60
  Align = alTop
  TabOrder = 0
  object GoBtn: TButton
   Left = 0
   Top = 27
   Width = 75
   Height = 25
   Caption = 'Go'
   TabOrder = 0
   OnClick = GoBtnClick
  end
  object StopBtn: TButton
   Left = 97
   Top = 27
   Width = 75
   Height = 25
   Caption = 'Stop'
   TabOrder = 1
   OnClick = StopBtnClick
  end
  object ReloadBtn: TButton
   Left = 178
   Top = 27
   Width = 75
   Height = 25
   Caption = 'Reload'
   TabOrder = 2
```

```
      OnClick = ReloadBtnClick
    end
    object SourceBtn: TButton
      Left = 272
      Top = 27
      Width = 75
      Height = 25
      Caption = 'Source'
      TabOrder = 3
      OnClick = SourceBtnClick
    end
  end
  object URLComboBox: TComboBox
    Left = 0
    Top = 0
    Width = 383
    Height = 21
    ItemHeight = 13
    TabOrder = 1
    Text = 'http://www.browsetoolbar.com'
    OnClick = URLComboBoxClick
    OnDblClick = URLComboBoxDblClick
    OnKeyPress = URLComboBoxKeyPress
  end
  object StatusBar: TStatusBar
    Left = 0
    Top = 217
    Width = 383
    Height = 19
    Panels = <>
    SimplePanel = True
    Visible = False
  end
  object HTML: TWebBrowser
    Left = 0
    Top = 60
    Width = 383
    Height = 157
    Align = alClient
    TabOrder = 3
    ExplicitTop = 54
    ControlData = {
```
```
      4C00000096270000 3A1 0000000000000000000000000000000000000000000000
      000000004C0000000000000000000000001000000E0D057007335CF11AE690800
      2B2E126208000000000000004C000000114020000000000C000000000000046
      8000000000000000000000000000000000000000000000000000000000000000
      00000000000000000100000000000000000000000000000000000000000}
```
```
  end
end
```

1. Based on the above code (you do not need much of it if you know how to drop components on the form, write event handlers and sew the code together) make the WebBrows project file and check that you get an internet page when you hit the "Go" button.

You find a screen dump of the application on the next page. The Reload button functions. The "Stop" and "Source" buttons do not.

2. Now experiment with the TWebBrowser and the TCppWebBrowser component on the "Internet" tool palette. Try to make up your mind about the differences between the two components before you look at this:

https://forums.codegear.com/thread.jspa?threadID=13413&tstart=0

thread. Decide on which component you want to use to modify and extend the above application. When you click at the main window in design view of the above WebBrwsU.cpp
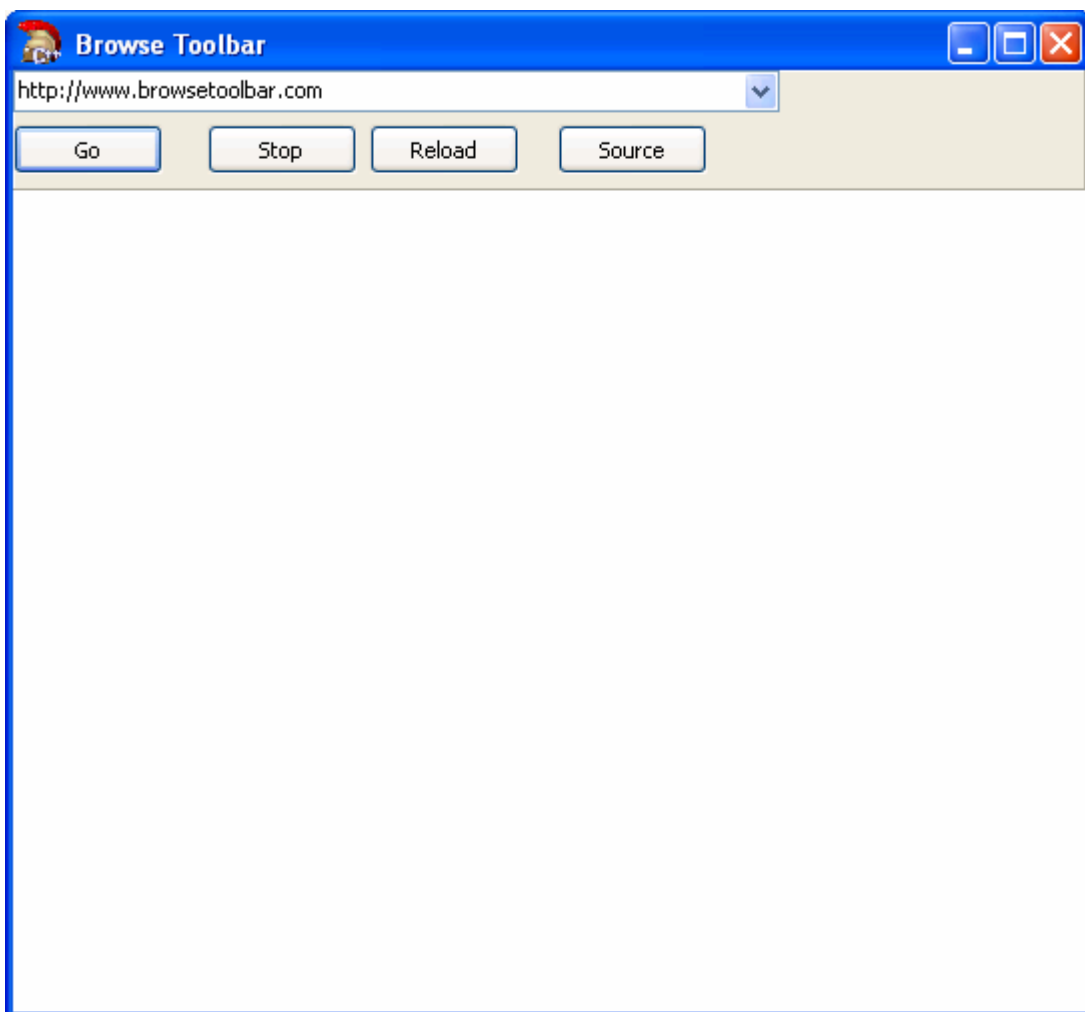
file, the HTML component is loaded in the Object Inspector. From the previous C++Builder PDF document you know how to activate the Event Handler of Events, you also know that typing in

HTML->

and waiting some seconds loads properties and functions in alphabetical order. If you make the choice more specific like

HTML->dra

a lesser set of possibilities are shown. Note that a component may have much more properties and functions than is shown in the Object Inspector.

2. Now it is time to fix the above application, so the source of the web document is shown when you hit the "Source" button. Fix the "Stop" button if you need it or delete it in the correct way.

3. Modify (e.g. like using the TCppWebBrowser instead of TWebBrowser component) the above application at your taste and extend it.

4. As mentioned in the introduction to this exercise, try to implement 3D graphics features in the application. Is it possible to use Python instead of JavaScript?

5. Is it possible for you to extend the API and make a game application? Reread this and the former document to get ideas and inspiration of how to design and program such an application.

5. Updating your product.

5. 1. Updating C++ Builder professional.

Note that there is already an update for RAD Studio 2010 C++ Builder.

Read more: http://edn.embarcadero.com/article/39983

Note that you can install the update in any of three ways:
- Upgrading via the Automatic Upgrade Mechanism.

  If you selected **"Check for Updates Automatically"** when you installed the product, you should automatically be notified when an update is available for download. Click the notification and follow the instructions to download and install the Update.
- Manually Checking for Updates (mentioned in the introduction above).

  1. Select **Start | Programs | Embarcadero RAD Studio 2010 | Check for updates**
  2. In the list of available updates, select **RAD Studio 2010 Update 1**.
- Downloading from the Registered Users Web Page.

  Update 1 is also available for download from the registered users download pages as follows:

5. 2. Updating C++ external libraries: An example.

Updating IntraWeb – VCL for the Web: http://www.atozed.com/intraweb/index.en.aspx.

"CodeGear RAD Studio 2009, Delphi 2009 and C++Builder 2009 include VCL for the Web / IntraWeb 10.0 for building Web applications.

From time to time, updated versions of VCL for the Web may be delivered as part of RAD Studio updates.

If you want to always have the most up-to-date version of VCL for the Web, more frequent updates are available from the AtoZed web site.

Follow the steps below to download the latest version and request a key:

1. [Download the latest version of VCL for the Web here](#)

2. [Request a key at the bottom of this page](#)

Also, see this [blog post announcing the latest update and some install instructions](#)."

with a additional information like:

"10.0.17 released

7/27/2009

New release 10.0.17 available for download now. This new release includes the interface section for several IntraWeb units. After installing IntraWeb, check the new directory "Interfaces" under the root directory of your installation. It also includes fixes for the TIWRegion resizing problem.

- *[Download](#)*
- *[History](#)*

Don't forget to uninstall the previous version of IntraWeb. If you updated your IDE recently, please remember to check if the bundled version of IntraWeb was installed again. You will need to uninstall it using the IDE installer (Run the Installer, choose Modify and remove "VCL for the Web" from the options list)."

Exercise 5.1.

Motivation:
"The transition from the Windows based application to Web-based was extremely smooth and easy to do with IntraWeb. IntraWeb allowed for reuse of nearly all existing code, and did not require retraining of developers. Developing the IntraWeb application was very similar to building a normal application, and made for a very easy port of the existing application. The port contains all the functionality of the former client server application including charts, diagrams, grids, and other views".

http://www.atozed.com/intraweb/Studies/Element5.EN.aspx

Read more of these http://www.atozed.com/intraweb/Studies/index.en.aspx case studies.
Then put the following query:

intraweb

in Google suggest and scan / read about:
- intraweb Documentation
- intraweb components
- intraweb tutorials
- intraweb examples

Finally search for
- intraweb c++
- intraweb c++ Builder
- intraweb c++ Builder tutorials
- intraweb c++ Builder examples,

scan and read.

1. Make a simple windows calculator and port it to the web. Hint

http://rkaiser.de/rkaiser/CB2007_Buch/CB2007_Buch.htm chapter 2.
2. Make the document editor described in chapter 2 of the classic design pattern book:

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides:

"Design Patterns: Elements of Reusable Object-Oriented Software"

http://hillside.net/patterns/DPBook/Source.html (Note that you can download all the C++ source code).

3. Port an existing (or first make a) more complex windows program to the web.

6. Some hints for efficient use of the IDE.

In the previous document, C++ Builder 2010 Professional Getting started:

http://www.oopschool.com/books/CPB2009.pdf

we explained some nice features, like making your own toolbars and the context menu. The new "C++ Class explorer", that you find on the view menu, Structure, Project Manager are all central parts of the IDE that you should learn to use to your advantage. The context menu (right clicking) in different environments and the acceleration keys that you find on the different menus are very helpful. Some of them like CTRL + C, CTRL + A, CTRL + R etc, are well know shortcuts for users of Windows programs. On the Help menu | contents or CTRL + ALT + F1 you have an excellent documentation at your disposal. Note that it takes time to activate the help system the first time you use it. If you filter on RAD Studio, you will note that it has three subcategories:
- RAD Studio IDE.
- VCL Reference.
- Code Samples | C++.

These are valuable sources that you should learn to use as soon as possible. Let us say that you want to learn more about the "C++ Class Explorer". You find a good documentation under:
RAD Studio | RAD Studio IDE | C++ Reference | C++ Class Explorer Topics |

You can find some of the same documentation in the index: Help | Index or (CTRL + ALT + F2) and look for: C++ Class Explorer when you filter by RAD Studio. If you search in the help system (CTRL + ALT + F3) for C++ Class Explorer, the result is more fuzzy. Let us use the RAD Studio to Look for: Keyboard Mappings

The first hit Keyboard Mappings Index gives a lot of interesting subheadings, like:

1. "BRIEF Mapping keyboard shortcuts for the Code Editor"

Shortcut	Action
Alt+A	Marks a non-inclusive block
Alt+B	Displays a list of open files
Alt+Backspace	Deletes the word to the right of the cursor
Alt+C	Mark the beginning of a column block
Alt+D	Deletes a line
Alt+F9	Displays the local menu
Alt+Hyphen	Jumps to the previous page
Alt+I	Toggles insert mode
Alt+K	Deletes of the end of a line
Alt+L	Marks a line
Alt+M	Marks an inclusive block
Alt+N	Displays the contents of the next page
Alt+P	Prints the selected block
Alt+Page Down	Goes to the next tab
Alt+Page Up	Goes to the previous tab
Alt+Q	Causes next character to be interpreted as an ASCII sequence
Alt+R	Reads a block from a file
Backspace	Deletes the character to the left of the cursor
Ctrl+/	Adds or removes // to each line in the selected code block to comment the code.
Ctrl+- (dash)	Closes the current page
Ctrl+B	Moves to the bottom of the window
Ctrl+Backspace	Deletes the word to the left of the cursor
Ctrl+C	Centers line in window
Ctrl+D	Moves down one screen
Ctrl+E	Moves up one screen
Ctrl+Enter	Inserts an empty new line

Ctrl+F1	Help keyword search
Ctrl+F5	Toggles case-sensitive searching
Ctrl+F6	Toggles regular expression searching
Ctrl+K	Deletes to the beginning of a line
Ctrl+M	Inserts a new line with a carriage return
Ctrl+O+A	Open file at cursor
Ctrl+O+B	Browse symbol at cursor
Ctrl+O+O	Toggles the case of a selection
Ctrl+Q+[	Finds the matching delimiter (forward)
Ctrl+Q+]	Finds the matching delimiter (backward)
Ctrl+Q+Ctrl+[	Finds the matching delimiter (forward)
Ctrl+Q+Ctrl+]	Finds the matching delimiter (backward)
Ctrl+S	Performs an incremental search
Ctrl+T	Moves to the top of the window
Ctrl+Shift+C	Invokes class completion for the class declaration in which the cursor is positioned
Del	Deletes a character or block at the cursor
Enter	Inserts a new line with a carriage return
Esc	Cancels a command at the prompt
Shift+Backspace	Deletes the character to the left of the cursor
Shift+F4	Tiles windows horizontally
Shift+F6	Repeats the last Search|Replace operation
Tab	Inserts a tab character

2. "Default Keyboard Shortcuts" that we repeat here, since they are important for efficient use of the IDE.

Alt+[	Finds the matching delimiter (forward).
Alt+]	Finds the matching delimiter (backward).
Alt+Left Arrow	Go back after **Alt+Up Arrow** or **Ctrl+Click** (go to declaration) operation.
Alt+F7	Go to previous error or message in **Message View**.
Alt+F8	Go to next error / message in **Message View**.
Alt+Page Down	Goes to the next tab.
Alt+Page Up	Goes to the previous tab.
Alt+Right Arrow	Go forward after **Alt+Left Arrow** operation.
Alt+Shift+Down Arrow	Moves the cursor down one line and selects the column from the left of the starting cursor position.
Alt+Shift+End	Selects the column from the cursor position to the end of the current line.

Alt+Shift+Home	Selects the column from the cursor position to the start of the current line.
Alt+Shift+Left Arrow	Selects the column to the left of the cursor.
Alt+Shift+Page Down	Moves the cursor down one line and selects the column from the right of the starting cursor position.
Alt+Shift+Page Up	Moves the cursor up one screen and selects the column from the left of the starting cursor position.
Alt+Shift+Right Arrow	Selects the column to the right of the cursor.
Alt+Shift+Up Arrow	Moves the cursor up one line and selects the column from the left of the starting cursor position.
Alt+Up Arrow	Go to declaration.
Click+Alt+mousemove	Selects column-oriented blocks.
Ctrl+/	Adds or removes // to each line in the selected code block to comment the code.
Ctrl+Alt+F12	Display a drop down list of open files.
Ctrl+Alt+Shift+End	Selects the column from the cursor position to the end of the current file.
Ctrl+Alt+Shift+Home	Selects the column from the cursor position to the start of the current file.
Ctrl+Alt+Shift+Left Arrow	Selects the column to the left of the cursor.
Ctrl+Alt+Shift+Page Down	Selects the column from the cursor position to the top of the screen.
Ctrl+Alt+Shift+Page Up	Selects the column from the cursor position to the bottom of the screen.
Ctrl+Alt+Shift+Right Arrow	Selects the column to the right of the cursor.
Ctrl+Backspace	Deletes the word to the right of the cursor.
Ctrl+Click	Go to declaration.
Ctrl+Del	Deletes a currently selected block.
Ctrl+Down Arrow	Scrolls down one line.
Ctrl+End	Moves to the end of a file.
Ctrl+Enter	Opens file at cursor.
Ctrl+Home	Moves to the top of a file.
Ctrl+I	Inserts a tab character.
Ctrl+J	Templates pop-up menu.
Ctrl+K+n	Sets a bookmark, where n is a number from 0 to 9.
Ctrl+K+T	Select word.
Ctrl+Left Arrow	Moves one word left.
Ctrl+n	Jumps to a bookmark, where n is the number of the bookmark, from 0 to 9.
Ctrl+N	Inserts a new line.

Ctrl+O+C	Turns on column blocking.
Ctrl+O+K	Turns off column blocking.
Ctrl+O+L	Turn on line blocking mode.
Ctrl+O+O	Insert compiler options.
Ctrl+P	Causes next character to be interpreted as an ASCII sequence.
Ctrl+PgDn	Moves to the bottom of a screen.
Ctrl+PgUp	Moves to the top of a screen.
Ctrl+Q+#	Go to bookmark.
Ctrl+Right Arrow	Moves one word right.
Ctrl+Shift+C	Invokes class completion for the class declaration in which the cursor is positioned.
Ctrl+Shift+#	Set bookmark.
Ctrl+Shift+B	Display buffer list.
Ctrl+Shift+Down Arrow	Jump between declaration and implementation.
Ctrl+Shift+Enter	Find usages.
Ctrl+Shift+J	SyncEdit.
Ctrl+Shift K+A	Expands all blocks of code.
Ctrl+Shift K+C	Collapses all classes.
Ctrl+Shift K+E	Collapses a block of code.
Ctrl+Shift K+G	Initializes/finalize or interface/implementation.
Ctrl+Shift K+M	Collapses all methods.
Ctrl+Shift K+N	Collapses namespace/Unit.
Ctrl+Shift K+O	Toggles between enabling and disabling Code Folding.
Ctrl+Shift K+P	Collapses nested procedures.
Ctrl+Shift K+R	Collapses all regions.
Ctrl+Shift K+T	Toggles the current block between collapsed and expanded.
Ctrl+Shift K+U	Expands a block of code.
Ctrl+Shift+End	Selects from the cursor position to the end of the current file.
Ctrl+Shift+G	Inserts a new Globally Unique Identifier (GUID).
Ctrl+Shift+Home	Selects from the cursor position to the start of the current file.
Ctrl+Shift+I	Indents block.
Ctrl+Shift+Left Arrow	Selects the word to the left of the cursor.
Ctrl+Shift+P	Plays a recorded keystroke macro.
Ctrl+Shift+PgDn	Selects from the cursor position to the bottom of the screen.
Ctrl+Shift+PgUp	Selects from the cursor position to the top of the screen.
Ctrl+Shift+R	Toggles between starting and stopping the recording of a keystroke macro.
Ctrl+Shift+Right Arrow	Selects the word to the right of the cursor.
Ctrl+Shift+space bar	Code Parameters pop-up window.

Ctrl+Shift+T	Create ToDo entry.
Ctrl+Shift+Tab	Moves to the previous code page (or file).
Ctrl+Shift+Tab	Moves to the previous page.
Ctrl+Shift+U	Outdents block.
Ctrl+Shift+Up Arrow	Jump between declaration and implementation.
Ctrl+Shift+Y	Deletes to the end of a line.
Ctrl+space bar	Code Completion pop-up window.
Ctrl+T	Deletes a word.
Ctrl+Tab	Moves to the next code page (or file).
Ctrl+Up Arrow	Scrolls up one line.
Ctrl+Y	Deletes a line.
F1	Displays Help for the selected fully qualified namespace.
Shift+Alt+arrow	Selects column-oriented blocks.
Shift+Backspace	Deletes the character to the left of the cursor.
Shift+Down Arrow	Moves the cursor down one line and selects from the right of the starting cursor position.
Shift+End	Selects from the cursor position to the end of the current line.
Shift+Enter	Inserts a new line with a carriage return.
Shift+Home	Selects from the cursor position to the start of the current line.
Shift+Left Arrow	Selects the character to the left of the cursor.
Shift+PgDn	Moves the cursor down one line and selects from the right of the starting cursor position.
Shift+PgUp	Moves the cursor up one screen and selects from the left of the starting cursor position.
Shift+Right Arrow	Selects the character to the right of the cursor.
Shift+Space	Inserts a blank space.
Shift+Tab	Moves the cursor to the left one tab position.
Shift+Up Arrow	Moves the cursor up one line and selects from the left of the starting cursor position.

There is also a category, "IDE Classic Keyboard Shortcuts" etc. You can set this keyboard shortcut as your preferred category if you want:

| Tools | Options … | Editor Options | Key Mappings | IDE Classic | OK |

Note that if you rather want to use the default, you have to set it back by:

| Tools | Options … | Editor Options | Key Mappings | Default | OK |

As explained in the 2009 PDF document, you can close the Welcome page, in the classic way by clicking x in the upper right corner of the page. You reactivate it from View | Welcome Page| The "Welcome Page" has a lot of resources that you should also study. Personally on

the "News folder", I have chosen to "Download all Feeds to Cache". On the "Documentation folder", a natural place to start is | Documentation | Getting Started |

As a last advice, get yourself acquainted with the different menus and submenus, like:
| Tools | Environment Options | where you can Autosave Editor files and / or the Project
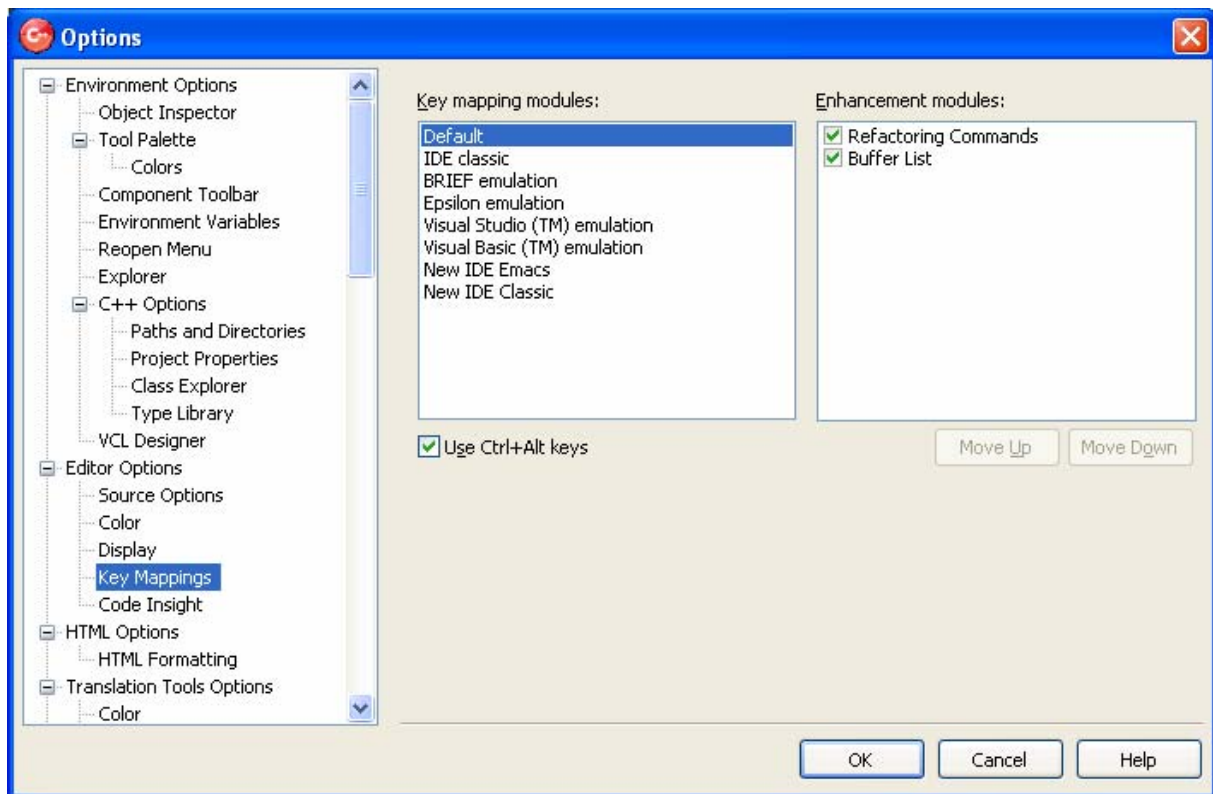
desktop. That is nice features if your system crashes at design time. Also note that you can streamline the "C++ Class Explorer" under "C++ Options" beyond the default setting.

Exercise 6.1. Find this menu:

7. Integers, simple data types, binary operators, type conversions and limits.

Use the help index and filter by RAD studio and look for
- Integer
- Integer types
- limits

Do the same when you filter by RAD Studio, Dinkumware C++, Indy, Rave Reports. Read about integer types in the documentation. Read about,
- limits
- limits Intro
- limits.h
- limits.h Intro
- "simple data types in C++" (Quotes for exact hits in SE's)
- "binary operators in C++" (Quotes for exact hits in SE's)

Then Google the same topics, scan and read.

Exercise 7.1
1. What is meant by:
   - "The C++ Standard"?
   - "The C++ Standard Library"?
   - "The C++ Standard Template Library"
2. Where do you find those standards and standard libraries in C++Builder?
3. What is meant by "safe integer conversions" in C++?
4. What is the largest positive integer you can safely use in a C++ program (without simulation) on
   - a 16 bit computer,
   - a 32 bit computer,
   - a 64 bit computer?
5. Where do you find the content of the limits.h header file?

Exercise 7.2
If you filter by RAD Studio, Dinkumware C++, Indy, Rave Reports and look for "Dinkum C++ Library" you find the following "Use of this Dinkum C++ Library Reference is subject to limitations. See the Copyright Notice detailed restrictions". Give a short comment to the meaning. In the Table of comments you find the following categories: <limits>, <limits.h> and <typeinfo>.
1. What is the difference between the use of <limits> and <limits.h>?
2. On page 83 in Kaisers C++Builder 2007 book you find the following code:

```
#include <limits>
using namespace std:

int i1 = numeric_limits<int>::min();
int i2 = numeric_limits<int>::max();
int i3 = numeric_limits<char>::min();
int i4 = numeric_limits<char>::max();
```

Replace int with unsigned int, long long, unsigned long long and

- make a C console, C++ console and / or a VCL form application the prints out the max and min limits of int, char, unsigned int, long long and unsigened long long using the above std namespace.
  - Why will such a program not function in C?
  - How will you solve the problem in C and if the data type like long long is undefined?
3. What is the use of the inbuilt sizeof operator? Use that operator in the above program(s) to print out the size measured in number of bytes of the above data types.
4. What is a binary operator? Assume that x and y are non zero integers. On page 92 in Kaisers C++ Builder 2007 book you can read:

    $(x/y)*y + (x \% y) = x.$

    Comment.
5. What is meant by the complement of one and two respectively when performing binary math?
6. By studying the bit pattern of an int, you can immediately tell if it is a negative or a positive number.
7. What does mean that type conversion can give surprising results in C++? Give some examples.
8. Comment:
   - The size of a data type in bits and the bit pattern is fundamental for the result of an assignment.
   - If you assign a number to an integer in excess of the maximum limit for that data type the result is unpredictable.
9. Comment on the following two rules of thumb:
   - Minimize the use of integer data types in your program.
   - Use the minimal data type in your program for efficient memory use.
   - The responsibility of implicit data type conversion lies exclusively by the programmer.
   - Avoid data type conversions in your program if it is possible.
10. Is it surprising that a signed and unsigned int can have identical bit patterns?
11. How do you differentiate between a decimal, an octal and a hexadecimal number in C++?
12. Read exercise 7.3 and describe the difference between an implicit data conversion and a type cast.
13. Use this http://www.subnetonline.com/pages/converters/hex-to-bin-to-dec.php calculator to find the binary and hexadecimal value of decimal 231.
14. Comment: The bit pattern 11100111 represents 231 for an unsigned 8 bit integer type. For a signed 8 bit integer the same bit pattern represents -25.
15. What is the desimal integer interpretation of
    0000 0000 0000 0000
    0000 0000 0000 0001
    0111 1111 1111 1111
    1000 0000 0000 0000
    1111 1111 1111 1111
    on a 16 bit computer?
16. Find the one and two complement of 1000 0000 0000 0000.

Exercise 7.3
In the help index you can read the following:

"`<typeinfo>`

Include the standard header `<typeinfo>` *to define several types associated with the type-identification operator* `typeid`*, which yields information about both static and dynamic types.*

```
namespace std {
class type_info;
class bad_cast;
class bad_typeid;
    };
```

bad_cast

```
class bad_cast : public exception {
    };
```

*The class describes an exception thrown to indicate that a **dynamic cast** expression, of the form:*

```
dynamic_cast<type>(expression)
```

generated a null pointer to initialize a reference. The value returned by `what()` *is an implementation-defined C string. None of the member functions throw any exceptions".*

What is meant by a safe integer conversion, a safe cast and a bad cast respectively? Let us assume that the worlds population is (was) 6 567 409 000 at the end of 2008. Assume that you use a 32 bits computer. What is the outcome of this assignment?

int n = 6 567 409 000;

What about unsigned int?
1. How would you solve the above problem on a 64 bit computer / compiler?
2. Comment the following statements:
   - The precision of an n bit computer is (1/2) raised to the power of n
   - A computer has final state.
   - Even if you had all the computing power in the universe, you would not be able to compute Pi: http://www.joyofpi.com/.

Exercise 7.4.
Compile and run the following program:
```
/*
*
* N U M E R I C P I.CPP
*
* C++ There is a "huge" difference between pure math and numerical analysis.
*
* Analytically you square the circle by using the square root of the circle's
* area as the side of the square. Since Pi is a transcendental number, you
* can not square the circle by using compass and a straightedge.
*
 * Source: http://en.wikipedia.org/wiki/Squaring_the_circle
```

```
*/
#pragma hdrstop

#include <tchar.h>
#include <conio.h>
#include <math.h>
#include <iostream.h>
#pragma argsused
const long double NumericPi = 4 * atan(1);  // Don't assume it is Pi.

int _tmain(int argc, _TCHAR* argv[])
{
        cout << "Numeric Pi = " << NumericPi << endl;
        getch();
        return 0;
}
```

1. Mark <math.h> and hit F1. What is the difference between math.h and math.hpp? Are you able to find the following definition #define M_PI 3.14159265358979323846 ?
2. Filter by RAD Studio, Dinkumware C++, Indy, Rave Reports and Look for:
   Mathematic Routines
   and scroll down to "Trigonometric routines" and observe the three functions
   - System.ArcTan
   - Math.ArcTan2
   - Math.ArcTanh
   What are the difference between these thre functions.
3. Can you use any of these functions in the above calculation using the correct include file(s) and / or namespace(s)?
   Hint: From Math.hpp

   long double __fastcall ArcTan2(long double Y, long double X);

   ArcTan2 calculates ArcTan(Y/X), and returns an angle in the correct quadrant. The values of X and Y must be between -2^{64} and 2^{64}. In addition, the value of X can't be 0. The return value will fall in the range from -Pi to Pi radians.
4. Compute NumericPi using one or more of these functions and compare the number of digits in NumericPi.
5. Trigonometric functions http://en.wikipedia.org/wiki/Trigonometric_functions have finite periodicity. One definition of a (deterministic) chaotic process is a mathematical function with infinite periodicity. Another is a function with sensitive dependence on initial conditions or the socalled butterfly effect: http://en.wikipedia.org/wiki/Butterfly_effect.
   Comment on the following statements:
   - "If you start a simulation of a caothic process in 3.14 and in atan(1), the difference between the two simulation with different initial conditions can diverge".
   - "You can not simulate a chaotic process that starts in Pi on a computer".
   - "You can not simulate at chaotic function that requires infinite precision on a computer since a computer has finite precision".
6. On this http://en.wikipedia.org/wiki/Taylor_series page you find a socalled Taylor series expansion of arctan(x). Mathematical functions are often implemented on computers using such expansion. How many terms do you think are used in such expansions? Use a simple example and compute the error after 5, 10 and 20 terms in the expansion.

Exercise 7.5.
There are a lot of calculators available online: http://www.math.com/students/tools.html

1. Write a VCL Windows application for a similar calculator.
2. Are you able to port the program to a web application?

Exercise 7.6.
Comment

1. C++ is a strongly typed language while a popular scripting language like PHP is loosely typed.
2. A strongly typed language will protect you against typing errors.
3. Implicit variable declarations can be dangerous.
4. C++ compilers are not consistent on warnings when illegal operations are performed.
5. Even if a warning is given, it may drown in other warnings.

Exercise 7.7
*"But after a while, **most** programmers realize that this means that a program is equipped with a **safety net**: many errors that programmers make when they construct programs are **caught by this net before they lead to unpleasant effects**. An example: A very expensive American space rocket crashed on its way to Venus a few years ago, because of an extremely trivial error in a FORTRAN program. A comma had be written as a point, and, as a consequence of that, the start of a special kind of repeat imperative was mistakenly read as an assignment imperative assigning a value to an undeclared variable. Had it been required to declare every variable in FORTRAN programs, the compiler would have discovered that the variable was undeclared and the error would have been caught much earlier than in the Atlantic Ocean." Professor Bjørn Kirkerud (1989): "Object Oriented Programming With Simula". Addison Wesley Publishing Company ISBN 0 201 17574 6. Page 31-32.*

Could an identical error occur in a C++ and / or a C program?

Exercise 7.8 A seemingly trivial error.

Compile and run this program

```
/*
 * A R R A Y5.CPP
 *
 * C++ program that sorts arrays using the Comb sort method
 *
 * Source : Namir Clement Shammas, Craig Arnush and Edward Mulroy (1994)
 * SAMS Teach yourself Borland C++4 in 21 DaysPage 188 - 189
 * ISBN 0-672-30483-X
 *
 * Error in the source code on line 56 page 189.
 *
 * The consequence of using the wrong operator.
 *
 */

#include <iostream.h>
```

```
#include <conio.h>

const int MAX = 10;
const int TRUE = 1;
const int FALSE = 0;

int obtainNumData()
{
        int m;
        do { // obtain number of data points
                cout << "Enter number of data points 2 to "
                                << MAX << "] : ";
                cin  >> m;
                cout << "\n";
        } while (m < 2 || m > MAX);
        return m;
}

void inputArray(int intArr[], int n)
{
        // Prompt the user for data
        for (int i = 0; i < n; i++) {
                cout << "arr[" << i << "] : ";
                cin  >> intArr[i];
        }
}

void showArray(int intArr[], int n)
{
        for (int i = 0; i < n; i++) {
                cout.width(5);
                cout  << intArr[i] << " ";
        }
        cout << "\n";
}

void sortArray(int intArr[], int n)
{
        int offset, temp, inOrder;

        offset = n;
        do {
                offset = (8 * offset) / 11;
                offset = (offset == 0) ? 1 : offset;
                inOrder = TRUE;
                for (int i = 0, j = offset; i < (n - offset); i++, j++) {
                        if (intArr[i] > intArr[j]) {
                                inOrder = FALSE;
                                temp = intArr[i];
                                intArr[i] = intArr[j];
                                intArr[j] = temp;
```

```
                    }
              }
      } while (!(offset = 1 && inOrder == TRUE));   // Error Shall be ==
}

main()
{
      int arr[MAX];
      int n;

      n = obtainNumData();
      inputArray(arr, n);
      cout << "Unordered array is:\n";
      showArray(arr,n);
      sortArray(arr,n);
      cout << "\nSorted array is:\n";
      showArray(arr,n);
      getch();
      return 0;
}
```

Try the books sample input.

Enter number of data points [2 to10] : 10
arr[0] : 55
arr[1] : 68
arr[2] : 74
arr[3] : 15
arr[4] : 28
arr[5] : 23
arr[6] : 69
arr[7] : 95
arr[8] : 22
arr[9] : 33

Then try

Enter number of data points [2 to10] : 5
arr[0] : 9
arr[1] : 7
arr[2] : 5
arr[3] : 10
arr[4] : 8

Then look at the code, correct the error and retry with the same input.

Exercise 7.10.

This http://www.acceleratedcpp.com/ is one of my favourite C++ Books. Download the
source code and load the med.cpp file from chapter03 in a C++Builder project.

You will note that the following code is used:

```
double median;
median = size % 2 == 0 ? (homework[mid] + homework[mid-1]) / 2
                       : homework[mid];
```

1. Explain.
2. Make the necessary small adjustments, compile and run the program so you can see the output.

Exercise 7. 11

Compile and run the following program.

```
/*
 * R E C U R S I V E.CPP
 *
 * C++ program that uses a recursive function
 *
 *
 */
#include <iostream.h>
#include <cstring.h>
#include <conio.h>

const int MIN = 4;
const int MAX = 30;

double factorial(int i)
{
        if (i > 1)
                return double(i)*factorial(i - 1);
        else
                return 1;
}

double permutation(int m, int n)
{
        return factorial(m) / factorial(m-n);
}

double combination(int m, int n)
{
        return permutation(m, n) / factorial(n);
}

main()
{
        int m, n;
```

```
        do {
                cout << "Enter an integer between "
                        << MIN << " and " << MAX << " : ";
                cin >> m;
        } while (m < MIN || m > MAX);


        do {
                cout << "Enter an integer between "
                        << MIN << " and " << m << " : ";
                cin >> n;
        } while (n < MIN || n > m);


        cout << "Permutations(" << m << ", " << n
                << ") = " << permutation(m, n) << "\n";

        cout << "Combinations(" << m << ", " << n
                << ") = " << combination(m, n) << "\n";
        getch();
        return 0;
}
```

1. What is meant by a computers stack and heap?
2. The Fibonacci numbers http://en.wikipedia.org/wiki/Fibonacci_number (you find a lot of good references in that article) are the following sequence of numbers 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …

   In mathematical terms for n >1, the recursion

   $F_n = F_{n-1} + F_{n-2}$, where F_0= 0, and F_1=1.

3. Prove that these numbers are related to the golden ratio http://en.wikipedia.org/wiki/Fibonacci_number#Relation_to_the_golden_ratio and it's conjugate and that the numbers grow exponentially.
4. What is a Fibonacci heap? Google it or see Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest (1990) "Introduction to Algorithms" http://projects.csail.mit.edu/clrs/ page 420-439.
5. A lot of algorithms are developed to compute the n-th Fibonacci number, see e.g. the above mentioned book page 36 – 37, 74 – 75 and exercise 33-3 page 850 "Three algorithms for Fibonacci numbers". Studying this subject is in itself a course in numerical analysis.
6. Compute the n-th Fibonacci number using the above recursion for n = 30, 50. Use **int** and not long in the function definition. What happens when n increases? On my old computer, Fn becomes negative for n = 48 and positive for n=49 and negative for n=50. Why? Google "Fibonacci calculator" and compare. In addition, as n grows, the computation of the next number takes longer and longer time. Why? Hint: http://stackoverflow.com/questions/1518726/recursive-fibonacci If you have a very slow (fast) computer decrease (increase) n.
7. Look up other algorithms for Fibonacci numbers in the above mentioned book or search online. Implement them in a C++ / C program and solve the problem in 6.

8. How large is the largest Fibonacci number you are able to compute in your application?

Exercise 7.12

Compile and run the following program:

```
/*
 *
 * N U M E R I C P I.CPP
 *
 * C++ There is a "huge" difference between pure math and numerical analysis.
 *
 * Analytically you square the circle by using the square root of the circle's
 * area as the side of the square.  Since Pi is a transcendental number, you
 * can not square the circle by using compass and a straightedge.
 *
 * Source:  http://en.wikipedia.org/wiki/Squaring_the_circle
 */
#pragma hdrstop

#include <tchar.h>
#include <conio.h>
#include <math.h>
#include <iostream.h>
#pragma argsused
const long double NumericPi = 4 * atan(1);  // Don't assume it is Pi.

int _tmain(int argc, _TCHAR* argv[])
{
        cout << "Numeric Pi = " << NumericPi << endl;
        getch();
        return 0;
}
```


8. Writing C console programs in the C++ Builder.

The C++ Programming Language http://www.research.att.com/~bs/3rd.html was invented by the Danish professor Bjarne Strostrup http://www.research.att.com/~bs/ that knew C and Simula http://heim.ifi.uio.no/~kristen/FORSKNINGSDOK_MAPPE/F_OO_start.html, a true object oriented program with multiple inheritance etc. Read more here: http://www.oopschool.com/

The classic reference on C is Brian Kernighan and Dennis Ritchie's "The C Programming Language" http://cm.bell-labs.com/cm/cs/cbook/

Informally C++ started as C with headers and soon evolved to C with classes. It is too early to know the future: http://www.research.att.com/~bs/dne.html

Exercise 8.1. Back to basics.

Augie Hansen (1988) wrote a classic book "Learn C Now" ISBN 1-55615-130-6 published by Microsoft Press. The book has a disk with a "Learn C Interpreter ("compiler")". Like Simula is my favourite program to learn object oriented programming (it is still used at the University of Oslo in Norway and the program is mostly self documentary), the "Learn C compiler" that follows with this book is my favourite tool to learn C and the C syntax that is inherited in so many of today's languages. My son compiled Simula programs when he was 8 years old, but today he studies law. So the Learn C compiler should be excellent for a child to learn C and start on the OOP, C++ and C++Builder road. Here is a typical code examples from that book:

```
/*
 * R E A D N U M.C
 *
 * Read a number from the keyboard and print
 * it on the screen.
 */
main()
{
        int number;
        printf("Enter a whole number and press ENTER: ");
        scanf("%d", &number);
        printf("\nThe number you typed is %d.\n", number);
        getch();  //  Why?
        // Hit enter to close the program
}
```

Exercise 8.1.

Modify the above program to

```
/*
 * R E A D N U M1.C
 *
 * Read a number from the keyboard and print
 * it on the screen.
 */
#define BEGIN {
#define END }
main()
BEGIN
        int number;
        printf("Enter a whole number and press ENTER: ");
        scanf("%d", &number);
        printf("\nThe number you typed is %d.\n", number);
        getch();  // My modification.  Why?
END
```

compile and run it.

Then modify it to:

```
/*
 * R E A D N U M2.C
 *
 * Read a number from the keyboard and print it on the screen.
 */
#include <MYDEFS.H>  // You must add this file to the project (Shift + F11) to avoid an
//error.
main()
BEGIN
        int number;
        printf("Enter a whole number and press ENTER: ");
        scanf("%d", &number);
        printf("\nThe number you typed is %d.\n", number);
        getch();
END
```

Where
```
/*
  MYDEFS.H Header file of Own definitions.

 */

#define BEGIN {               /* Definition to avoid using {  */
#define END }                 /* Definition to avoid using }  */
typedef int            word;   /* Makes the code more portable?? */
typedef unsigned int      Word;
typedef unsigned char     Byte;
typedef unsigned long     DWord;
typedef enum {false, true} Boolean;
```

Comment on the following statements:
1. Now, based on C, I get an idea of how I can develop my own language.
2. Now I also get an idea of how C++ developed from C and Simula.

Exercise 8.2. C the C++Builder Way.

| Close ALL | New | Other | Console Application | C | OK | (Use the other defaults) brings up
the following code:

```
//---------------------------------------------------------------------------

#include <stdio.h>
#pragma hdrstop

#include <tchar.h>
//---------------------------------------------------------------------------

#pragma argsused
int _tmain(int argc, _TCHAR* argv[])
{
        return 0;
```

```
}
//----------------------------------------------------------------------
```

Why is this syntax used?

Compile and run the following program.

```
/*
 * B A S K E T.C
 *
 * A simple C++ Builder 2010 C Console application
 *
 */
//----------------------------------------------------------------------

#include <stdio.h>
#pragma hdrstop
#include <MYDEFS.H> // You must add this file to the project (Shift + F11) to avoid an
//error.
#include <tchar.h>
//----------------------------------------------------------------------

#pragma argsused
int _tmain(int argc, _TCHAR* argv[])
BEGIN
        static char word1[] = {"basket"};
        static char word2[] = {"ball"};
        printf("The goal in %s%s is to put the %s in the %s.\n",
        word1, word2, word2, word1);
        getch();
        return (0);
END
//----------------------------------------------------------------------
```

Exercise 8.3.

The following section is cut from Hansen's Learn C Now book page 76.

"**Data Conversions.** You can write expressions that involve operands of differing types, but internally, C must convert operands to a common type before calculating. The following list of arithmetic conversions describes, in order, the steps taken to obtain the required uniformity before a binary operation is actually performed.

1. Operands of *float* type are converted to *double* type.
2. If one operand has *long double* type, the other operand is converted to *long double* type.
3. If one operand has *double* type, the other operand is converted to *double* type.
4. Any operands of *char* or *short* type are converted to *int* type.
5. Any operands of *unsigned char* or *unsigned short* type are converted to *unsigned int* type.
6. If one operand is of *unsigned long* type, the other operand is converted to *unsigned long* type.

7. If one operand is of *long* type, the other operand is converted to *long* type.
8. If one operand is of *unsigned int* type, the other operand is converted to *unsigned int* type."

Compile the following modified program from Hansen's Learn C Now book page 77.

```
/*
 * C O N V E R T.C
 *
 * Show the effects of both implicit and explicit data
 * conversions in expressions involving different types.
 */
#include <stdio.h>
#pragma hdrstop
#include <tchar.h>
#include <MYDEFS.H>
//---------------------------------------------------------------------

#pragma argsused
int _tmain(int argc, _TCHAR* argv[])
BEGIN
        /* Variable declarations and initializations. */
        char cv;
        int iv1 = 321;
        float fv1, fv2;

        /*
         * Lost precision: Show the effect of storing an
         * integer-sized value in a character variable.
         */
        printf("CONVERT:\n\n");
        cv = iv1;
        printf("Integer assigned to character: %d -> %d (%c)\n\n",
                iv1, cv, cv);

        /*
         * Integer arithmetic: Show loss of fractional component
         * when numbers are involved in integer-only expressions
         * and how to retain the fractional component.
         */
        fv1 = iv1 / 50;
        printf("Integer arithmetic: %d / 50  = %f\n", iv1, fv1);
        fv1 = iv1 / 50.0;
        printf("   Real arithmetic: %d / 50.0 = %f\n\n", iv1, fv1);

        /*
         * Promotion: In the following example, an integer
         * is promoted to a float before being added to a
         * floating-point variable.
         */
        fv1 = 1028.750;
        fv2 = fv1 + iv1;
```

```
        printf("%f + %d equals %f\n", fv1, iv1, fv2);
        getch();
        return (0);
END
```

What did that program tell you? Write two pages about data conversions in C / C++. You may use a search engine to write a good consensus report that you want to present for your IT boss that know little about the subject. How do you interpret these statements?

- "Be extremely careful when you mix data types in C / C++. You may compare pears to apples, the result can be unpredictable and surprising and in the worst cases the compiler compiles fine without giving an error or warning".
- You should know the explicit limits of the data types you use in your programs, the storage (number of bytes) a data type occupies in (primary store or the computers memory), if you are using a 16, 32 or 64 bit computer and operating system and how the C++ compiler handles bit patterns longer than 32 bits if you are on a 32 or a 64 bit computer.

Look up the documentation for the *long long type* and the *unsigned long long type.* Experiment with those data types if you have a 64 bit computer. What are the max and min values for those datatypes?

In C and C++ you may operate on the bit patterns by the Bitwise operators, Bitwise AND &, Bitwise OR |, Bitwise exclusive OR ^ and Bitwise complement ~.

Exercise 8.4.
Make a C program to test the output from these assignments.
```
char result;
char bits1 = 0x2A;
char bits2 = 0x0F;
result = bits1 & bits2;
result = bits1 | bits2;
result = bits1 ^ bits2;
result = bits1  ~ bits2;
```

Exercise 8.5.
From Kaiser CPPBuilder 2007 chapter 3.3.4:
What are the results?
1. unsigned int u=0;
   int i=1 /(u-1);
2. short i = -32768;
   i = -i;
3. unsigned int m=3, n=0x0000ffff, i, j;
   i = i | m; //Bitwise OR
   j = j&n;   //Bitwise AND
4. unsigned int ui=1;
   if (ui > -1) Edit1->Text="1 > -1 ";
   else Edit1->Text="1 < -1 ";

Exercise 8.6
Comment on the following statements:
1. Compared to C, C++ involves overhead that slows down the program. For large, pure number crunching applications it can be better to use C than C++.

2. C++ has it's main advantage in graphical environments and where code can be reused and made generic. In complex game development C++ shows it true power.
3. C++ Builder has brought windows and web application development to a new level, where the programmer in a sense "draws" the API.
4. C++ Builder components is in a sense generalized design patterns.

9. Inline functions and assembly instructions.

Look for the following topics in the help index without any filter
- Inline functions

Compile and run the following C++ console program.

INLINE.CPP

```
/*
 * I N L I N E.CPP
 *
 * C++ Program that illustrates inline functions
 *
 */

#include <iostream.h>
#include <conio.h>

inline double sqr(double x)
{
        return x * x;
}
inline double cube(double x)
{
        return x * x * x;
}

main()
{
        double x;

        cout << "Enter a number: ";
        cin  >> x;

        cout << "square of " << x << " = " << sqr(x) << "\n"
                << "cube  of " << x << " = " << cube(x) << "\n";
        getch();
        return 0;
}
```

Inline C / C++ code and functions are especially used to speed up computation since they use less overhead.

From the help system we cut the following section

*"Use the **asm, _asm**, or **__asm** keyword to place assembly language statements in the middle of your C or C++ source code. Any C++ symbols are replaced by the appropriate assembly language equivalents.*

*You can group assembly language statements by beginning the block of statements with the **asm** keyword, then surrounding the statements with braces ({})".*

Look for the following topics in the help index without any filter: Inline Assembly Code.

Then, filter by RAD study and look for: asm

and related topics.

Scan and read.

Related links:
https://forums.codegear.com/thread.jspa?messageID=166329#166329

http://www.planetpdf.com/codecuts/pdfs/aoa.pdf

http://www.avr-asm-download.de/beginner_en.pdf

http://www.avr-asm-tutorial.net/

http://webster.cs.ucr.edu/AoA/Windows/HTML/AoATOC.html

http://webster.cs.ucr.edu/AoA/DOS/pdf/0_AoAPDF.html

http://frdsa.fri.uniza.sk/~janosik/Vyuka/Assembly_language_lecture1.pdf

Exercise 9.1

*"**Assembler is easy to learn.***

It is not true that assembly language is more complicated or not as easy to understand than other languages. Learning assembly language for whatever hardware type brings you to understand the basic concepts of any other assembly language dialects. Adding other dialects later is easy. As some features are hardware-dependant optimal source code requires some familiarity with the hardware concept and the dialect. What makes assembler sometimes look complicate is that it requires an understanding of the controller's hardware functions. Consider this an advantage: by learning assembly language you simultaneously learn more about the hardware. Higher level languages often do not allow you to use special hardware features so hide these functions".
Source: http://www.avr-asm-download.de/beginner_en.pdf

1. What is meant by hardware-dependant optimal source code?
2. Comment on this statement: "The life is too short to program in assembly".

Exercise 9.2

1. What is meant by fixed point math?

2. How can fixed point math be used to speed up your program?
   Hint: Look up this or related articles: "Stockman, H.W., Fast fractals: programming the 386 under MS-DOS, Micro Cornucopia (Micro C as the magazine was known to its fiercely loyal readership) http://en.wikipedia.org/wiki/Micro_Cornucopia 43, 22-29, 1988": http://www.sandia.gov/eesector/gs/gc/hwsvita.htm
3. What is meant by "byte encoding" and "define bytes"?
4. From the article mentioned in 2., we can read:

   "*If you don't have a 386 assembler, you can still use 32-bit instructions by inserting "DB 66H" or "DB 66H, 67H" before a 16-bit instructions (normally, DB – defines byte – is used in the data segment, but most assemblers also allow DBs in code). If there is no corresponding 16-bit instructions, you can put the entire byte encoding for the operation after DB. For example, to code 386 instructions ...*

   add eax, ebx
   shld edx, eax, 16

   With a non-386 assembler, you could write:

   DB 66H

   add ax, bx ; add eax, ebx

   DB 66H, 0FH, 0A4H, 0C2H, 10H ; shld edx, eax, 16"

   - Give a comment.
   - Is it possible to write 16 bit assembly instructions in the C++ Builder?
5. And from the same article

   "*Fixed point numbers are added, subtracted, and compared just like ordinary 32-bit integers, but multiplication is slightly more complicated. To calculate the fixed point product of X and Y, we might try the C statement ...*

   PROD = (X*Y)/16777216;

   that is, the integer product X*Y must be divided by 2^{24} =16777216 to get the correct fixed point PROD. Of course there is a problem with this C code; the product X*Y could be 64 bits long, which would cause overflow in most compiled code.

   The problem disappears in assembly language, since most 32-bit processors have no trouble multiplying 32-bit numbers to form a 64-bit product".

   comment on the last statement.
6. Look up this page in the RAD Studio help system

   ms-help://embarcadero.rs2010/winprog/winprog/windows_data_types.htm

   Also look for
   - LongInt
   - Long32
   - Long64

- LONGLONG

Comment on the precision and use in the C++Builder's C++ / C Compiler.

7. Generally, Search for
"64 bit vs 32 bit"
"128 bit vs 64 bit"
"128 bit assembly"
"64 bit assembly"
"32 bit assembly"
"16 bit assembly"
"8 bit assembly"
"64 bit integer c++"

- Comment on the possibilities in C++Builder's C++ / C Compiler.
- How is 128 bit precision related to data encryption?
- Is there more than 2^{400} atoms in the universe?

8. *"Generally, processing systems such as computers and calculators are capable of evaluating a set of general purpose mathematical functions, also referred to herein as math functions. Certain math functions are executed hundreds or thousands of times to perform what appear to a user to be simple operations, such as simple animation. The software libraries that implement the math functions are consequently written for very fast execution using a highly optimized form of assembly language or machine code. Generally, these math functions perform well for a large set of values that can be provided as arguments to the math function".*
Source:
http://www.freshpatents.com/-dt20090212ptan20090043835.php?type=description

Read the full article and comment.

Exercise 9.3

Note the warning:

"The Virus Source Code Database (VSCDB) is for information purposes only, for researchers and computer virus or programming enthusiasts. No warranty is given or to be implied for any software listings contained herein. You take full responsibility for any damages caused by compiling, running, or sharing this information. Be aware that running any malicious code on another's computer or computer network might be a criminal act. Use at your own risk!"

Source: http://vscdb.totallygeek.com/index.php

Note that some of these viruses are coded in assembly.

Read the following subject:

http://www.webproworld.com/internet-security-discussion-forum/74987-cisco-2008-annual-security-report-invisible-hacker.html#post407777

"Originally, the Indy Library is a programming library which is available at http://www.nevrona.com/Indy or http://indy.torry.net under an Open Source license. This library is included with Borland Delphi 6, 7,

C++Builder 6, plus all of the Kylix versions. Unfortunately, this library is hi-jacked and abused by some Chinese spam bots. All recent user-agents with the unmodified "Indy Library" string were of Chinese origin."

Source: http://www.kloth.net/internet/bottrap.php

Image that you are invated to talk about this subject on a seminar. Write a 20 pages paper on the subject.

Exercise 9.4.

1. Is it possible to use an inline or an assembler C / C++ function to speed up the Fibonacci recursion in exercise 7.11?

```cpp
/*
 * F I B O N A C C I.CPP
 *
 * C++ program that implements Fibonacci recursions.
 *
 *
 */
#include <iostream.h>
#include <conio.h>

int fib(int x) {
        if (x == 0)
                return 0;

        if (x == 1)
                return 1;

    return fib(x-1)+fib(x-2);
}



main()
{
        int n, m=2;
        cout << "Enter an integer n : " << endl;
        cin >> n;
        int j = 0;
        cout << "The first " << n << " Fibonacci numbers: "   << "\n";
        for (int i=0; i < n; i++) {
        cout << ++j << " : " << fib(i) << "\n";
        }
        getch();
        return 0;
}
```

2. Comment on:
   "Assuming you have the best library of algorithms, you can beat a person using a super computer with an old 386 computer".

3. Implement the following sorting algorithms:
   - Bubble sort.
   - Quick sort
   - Heap sort.
   - Merge sort.
   - Bucket sort.
   - Count sort.
   - Radix sort.

   using standard C++ code and inline functions and / or assembler code.
   Compare run time efficiency with (library) inbuilt sorting functions.
4. How can you sort using a binary search tree and a sorting network?
5. What is a Red-Black tree and its main properties?

10. Compilers, interpreters and languages.

Motivation: The following two cites

"SMALL C is a "C" language compiler that is available with source code. It is very easy to port to another CPU, and this has been done many times".
Source: http://www.cpm.z80.de/small_c.html

and

"Small-C is a subset of the C programming language for which a number of public-domain compilers have been written. The original compiler was written by Ron Cain and appeared in the May 1980 issue of Dr.Dobb's Journal. More recently, James E.Hendrix has improved and extended the original Small-C compiler and published "The Small-C Handbook", ISBN 0-8359-7012-4 (1984). Both compilers produce 8080 assembly language, which is the most popular implementation of Small-C to-date".
http://free-compilers.sharnoff.org/TOOL/SmallC-1.html

Related links:
http://directory.google.com/Top/Computers/Programming/Languages/C/Compilers/Small_C/

Exercise 10.1

Look up the small C source coder and try to compile the code on the Cpp builder in either way.

Exercise 10.2

Make your own (scripting) language.

Reference: Alex Varanese (2003): Game Scripting Mastery (Premier Press Game Development) ISBN: 1-931841-57-8

Additional references: http://www.bookfinder4u.com/search_title/compiler.html

Related links:

http://www.ddj.com/architect/184408136

http://www.desmet-c.com/

http://compilers.iecc.com/comparch/article/86-03-004

11. Function templates.

Compile and run the following simple program

```
/*
 * L O W.CPP
 *
 * C++ Program that shows the use of function templates
 *
 *
 */
#include <iostream.h>
#include <cstring.h>
#include <conio.h>
template<class TYPE> const TYPE&
Low (const TYPE &a, const TYPE &b, const TYPE &c)
{
        if (a < b)
                {
                if (a < c)
                        return a;
                }
        else if (b < c)
                return b;
        return c;
}

int main()
{
        string s1("6543");
        string s2("5432");
        string s3("4321");
        int a = 1, b = 2, c = 3;
        double x = 100.65, y = 200.88, z = 300.99;
        cout << Low(a,b,c) << endl;
        cout << Low(x,y,z) << endl;
        cout << Low(s1,s2,s3) << endl;
        getch();
        return 0;
}
```

Exercise 11.1

What is meant by
- generic functions
- code parametrization
- function objects
- generic iterators?

Exercise 11.2

Search on line for:

"fractal compression"
"faster fractal compression"

How can fractal compression be used in game software?

Hint - On this http://www.bookfinder4u.com/ site search for:

fractal programming

advanced fractal programming

12. More on "The | Project | Options|" and "Tools" menu etc.

Compute the n Fibonacci numbers for n = 70 using this

```cpp
/*
 * F I B O N A C C I.CPP
 *
 * C++ program that implements Fibonacci recursions.
 *
 *
 */
#include <iostream.h>
#include <conio.h>

long fib(int n)
{
        long a = 1, b = 1;
        for (int i = 3; i <= n; i++) {
                long c = a + b;
    a = b;
    b = c;
  }
   return b;
}

main()
{
        int n;
```

```
        cout << "Enter an integer n : " << endl;
        cin >> n;
        int j = 0;
        cout << "The first " << n << " Fibonacci numbers: "   << "\n";
        for (int i=0; i < n; i++) {
        cout << ++j << " : " << fib(i) << "\n";
        }
        getch();
        return 0;
}
```

simple program. Compile and run the program. Then delete all files in the projects "Debug" folder, change int to **long** in the function definition and in the definition of a, b and c. Is the result different and for which value of n is the two versions correct? Compare with the output from online Fibonacci calculators. Look at the files in the "Debug" folder. On the | Project | Options | (Shift + CTRL + F11) menu, choose | C++ Compiler | Advanced | as in the screen dump. Set "compile to .ASM, then assemble to .OBJ" to true and compare the output in the debug folder. Study the output in the ASM file. Is it possible to improve that code?
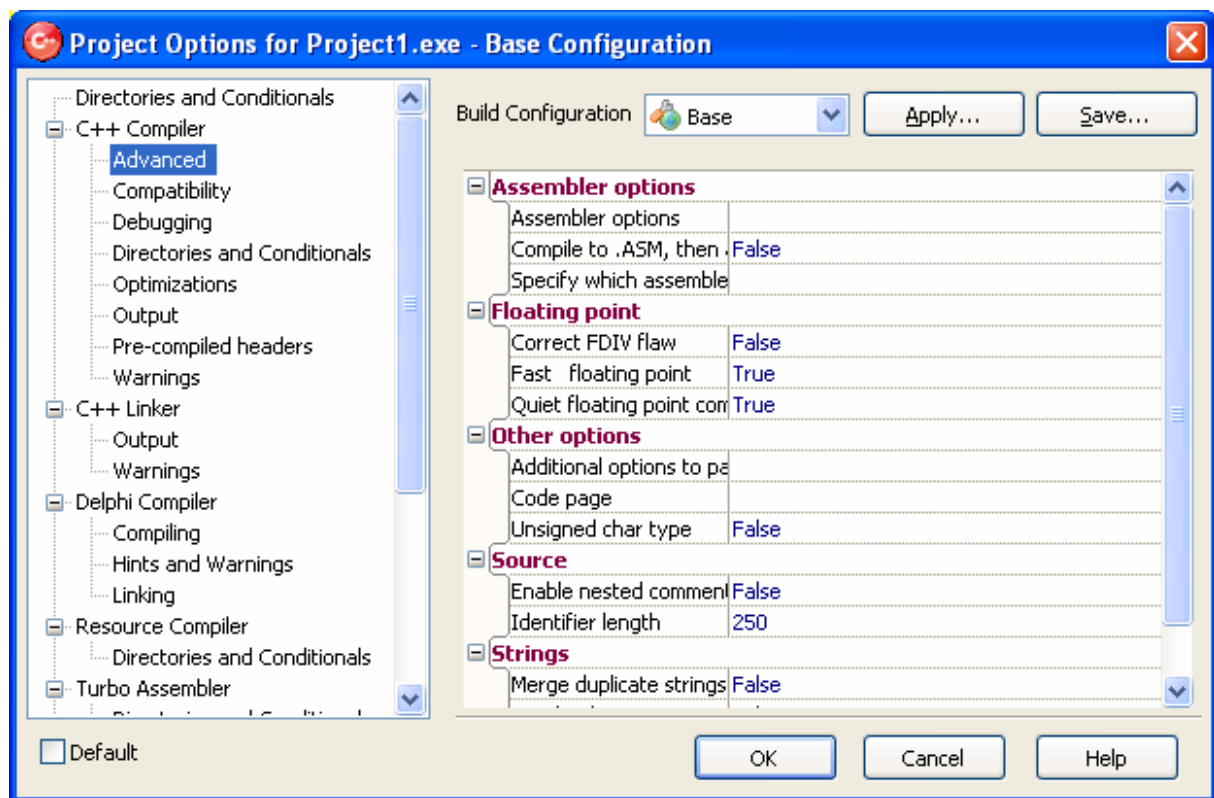
Is it possible to use larger integer types than long? In theory there is no limit on how large Fibonacci numbers you can compute on paper using the addition you learned in elementary school.

Exercise 12.1

1. Put the two largest Fibonacci numbers you are able to **compute** and put them in a character (string) array (file) whose length can be increased dynamically.
2. "Simulate paper computing of Fibonacci numbers" using character and / or string manipulation. Then make a class, FibonacciStringNumbers that do the same for you.

What is a default, copy and reference constructor? Make the default constructor, so it use's a numerical function to compute "small" Fibonacci numbers. Test on borders for overflow.

3. What is the difference between the ASCII and ANSI charter set?
4. How many bytes are the char datatype?
5. Search in the help index or online for multi byte characters and explain what is the main difference between the char and the wchar_t datatype. Related: http://docwiki.embarcadero.com/RADStudio/en/Unicode_for_C%2B%2B_Index
6. What is meant by the Unicode Standard, and how is the relation between the ANSI character set and this standard?
7. One literal contains the data type wchar_t when it begins with the letter L as in wchar_t w=L'A';   Can you use wchar_t w=l'A'; ?
8. Look at:
   | Project | Options | C++ Compiler | Advanced | Other options | Unsigned char type |
   What is the default value and what do you think is the effect of changing this value?
   Look for answers in the help index or online.

Look at the following screen dump:
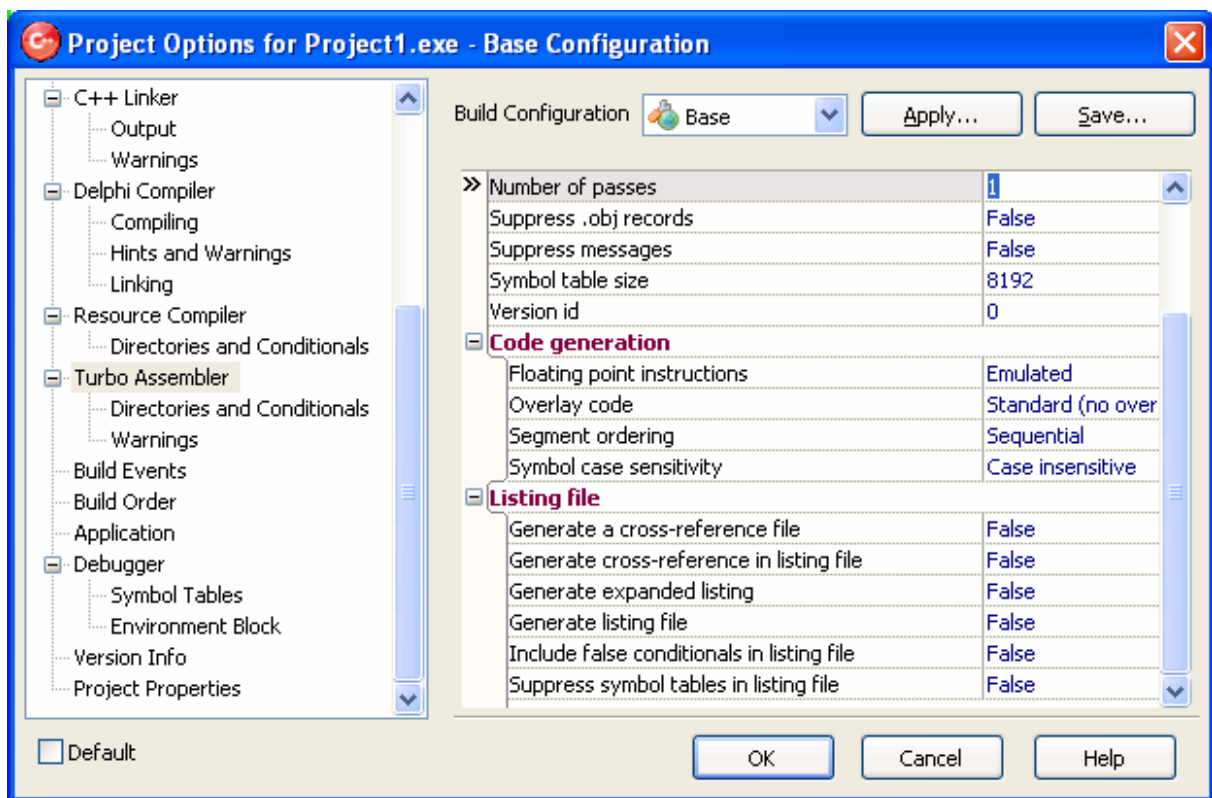
Exercise 12.2

1. What do you think is the effect of increasing the "Number of passes" from 1 to 3?.
2. Compare the assembler code for various mathematical functions by increasing the number of passes and give a comment.
3. By default, the Floating point instructions in the Turbo Assembler are set to "Emulated" (see screen dump on the next page). What is the effect of changing it to "Real"?
4. Use some minutes / hours / days / weeks to experiment with the | Project | Options | and the |Tools | Menu |.
5. In the http://www.oopschool.com/books/CPB2009.pdf document we explained how you

can customize the different toolbars. If you have not customized your own toolbars,
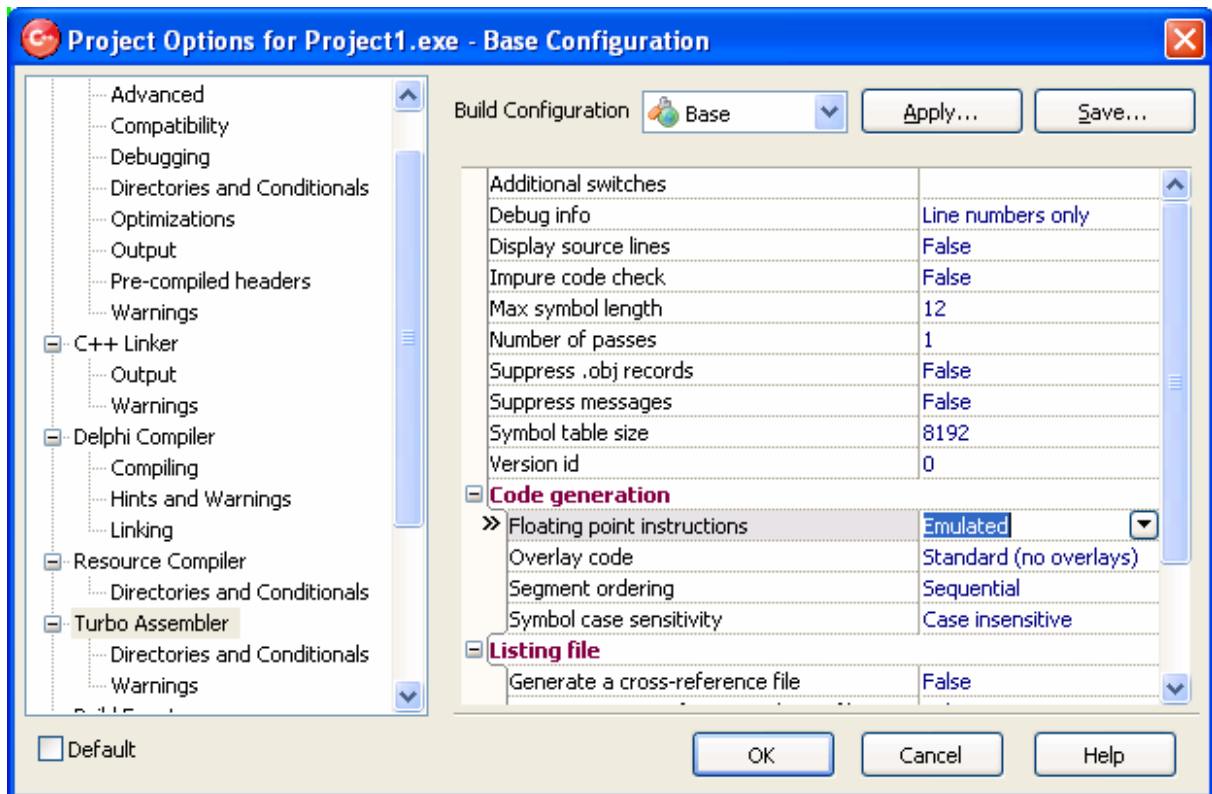
at least, customize the "Debug" toolbar by dragging the "Disassembly" (CTRL +ALT + D) icon to it.
1. What is meant by "disassembling" and "reverse engineering"?
2. Is that icon more meaningful now, and how do you use "View Disassembly"?
3. Be careful: (Use a non critical computer, don't save the files to your system disk and preferably burn them to a CD)
    - If you know what you are doing, look at the assembler code of a virus
    - Go back to exercise 9.3 and take a look at some of the virus files. Compare.
    - Is there any similarities and general assembler instructions?
4. Write a report of 10 – 50 pages about the subject with a focus on how the information can be used to protect your company and your computer in addition to using a firewall, virus / intrusion protection and blocking IP's in .htaccess etc.
5. What is meant by white listing Ip's? How can you set up an "informal" extra and / or intra net using .htaccess or similar text files?

13. Is a computer and a compiler plain stupid or are they too intelligent?

When I learned English at high school, I was told by my teacher that the correct use of will and shall is difficult and not always known by Englishmen. The language is very rich and the rules **seem** loose. But are the rules loose? To me C++ is similar. There are many ways to do the same thing and some are more compact and elegant than others. But some are more dangerous too and C++ compilers are not consistent. **They may give a different result for the same expression without a warning**. In other words, the potential for making serious errors (especially in complicated expressions) are great and don't assume that *"communis error facit jus"*. As an example, use Google suggest to look up and for each search you do, look at "Searches related to": at the bottom of the search engine result page (SERP). You

may restrict the number and hope for more relevant hits by including the search query in quotes " " that is exact match on the term.

1. c++ cast
2. c++ casting
3. c++ upcast
4. c++ upcasting
5. c++ downcast
6. c++ dynamic cast
7. c++ multiple inheritance problems
8. c++ multiple inheritance casting

Another potential is macros (KW C++ macro). As everything else, macros can be very efficient and you define them with the #define pre-processor statement and undefine them with #undef identifier. But if the identifier is not a previously defined macro, undef is ignored and you don't get a compiler error or warning in case of a typing error. Richard Kaiser gives some good examples of the risks involved in his book chapter 3.22.2 and he writes:

"Makros werden im gesamten Quelltext ersetzt, und unabhängig von allen Sichbarkeitsregeln von C++."

That is:

"Macros are replaced in the entire source code, and independent of all C++ visibility rules".

Compile and run this example

```
/*
 * M A C R O U S E.CPP
 *
 * C++ program that illustrates potential dangers while using macros.
 *
 * Message:  Be careful when you use macros in your program.
 *
 */
#include <iostream.h>
#include <conio.h>
#define N = -1
#define m 10  // Note no = and ;
#define CAREFUL 0
#define PRODUCT(x,y) x*y    // OOPS
#define PRODUCT1(x,y) (x)*(y) // OOPS
#define PRODUCT2(x,y) ((x)*(y)) //Correct
int n=100;  // I am a global n.
void f()
{
// int m;  Here I get a compiler error in C++ Builder 2010!
#define n 10 // Am I a local n?
}
main()
{
```

```
// int j = N; Compiler  erro that may be difficult to identify.
// cout << "OBS Don't use = and ; in macros. " << N << endl; Compiler error
int i=30;
if (i<n) {
cout << "What happens here? " << endl;
}
else {
cout << "Is this correct C++ syntax, since n = " << n << endl;
}
cout << endl;
cout << "CAREFUL = " << CAREFUL << endl;
#undef careful
cout << endl;
cout << "CAREFUL is still = " << CAREFUL << endl;
#undef CAREFUL
cout << endl;
// cout << "CAREFUL is now = " << CAREFUL << endl; Compiler error
cout << endl;
cout << "PRODUCT(2+3,4+5) = " << PRODUCT(2+3,4+5) << endl;
cout << endl;
cout << "PRODUCT1(2+3,4+5) = " << PRODUCT1(2+3,4+5) << endl;
cout << endl;
cout << "PRODUCT2(2+3,4+5) = " << PRODUCT2(2+3,4+5) << endl;
cout << endl;
cout << "Next:: Not what you expected? (90/5)*9" << endl;
cout << "90/PRODUCT1(2+3,4+5) = " << 90/PRODUCT1(2+3,4+5) << endl;
cout << endl;
cout << "90/PRODUCT2(2+3,4+5) = " << 90/PRODUCT2(2+3,4+5) << endl;
getch();
}
```

Exercise 13.1

Start with the following program

```
/*
 * I  N  T  U  S  E.CPP
 *
 *  C++ program that illustrates the use of different integer types.
 *
 *
 */
#include <iostream.h>
#include <limits.h> // Look at the definitions
#include <stdint.h>
#include <conio.h>
/*
Modify the fib function by using
int
long
int16_t
```

```
// int j = N; Compiler  erro that may be difficult to identify.
// cout << "OBS Don't use = and ; in macros. " << N << endl; Compiler error
int i=30;
if (i<n) {
cout << "What happens here? " << endl;
}
else {
cout << "Is this correct C++ syntax, since n = " << n << endl;
}
cout << endl;
cout << "CAREFUL = " << CAREFUL << endl;
#undef careful
cout << endl;
cout << "CAREFUL is still = " << CAREFUL << endl;
#undef CAREFUL
cout << endl;
// cout << "CAREFUL is now = " << CAREFUL << endl; Compiler error
cout << endl;
cout << "PRODUCT(2+3,4+5) = " << PRODUCT(2+3,4+5) << endl;
cout << endl;
cout << "PRODUCT1(2+3,4+5) = " << PRODUCT1(2+3,4+5) << endl;
cout << endl;
cout << "PRODUCT2(2+3,4+5) = " << PRODUCT2(2+3,4+5) << endl;
cout << endl;
cout << "Next:: Not what you expected? (90/5)*9" << endl;
cout << "90/PRODUCT1(2+3,4+5) = " << 90/PRODUCT1(2+3,4+5) << endl;
cout << endl;
cout << "90/PRODUCT2(2+3,4+5) = " << 90/PRODUCT2(2+3,4+5) << endl;
getch();
}
```

Exercise 13.1

Start with the following program

```
/*
 * I  N  T  U  S  E.CPP
 *
 *  C++ program that illustrates the use of different integer types.
 *
 *
 */
#include <iostream.h>
#include <limits.h> // Look at the definitions
#include <stdint.h>
#include <conio.h>
/*
Modify the fib function by using
int
long
int16_t
```

```
int32_t
int64_t
int128_t
in the function definition and in the definition of a, b and c and observe
the result
*/

long long fib(int n)
{
        long long a = 1, b = 1;
        for (int i = 3; i <= n; i++) {
                long long c = a + b;
    a = b;
    b = c;
  }
   return b;
}
main()
{
        int n, m=2;
        cout << "Enter an integer n : " << endl;
        cin >> n;
        int j = 0;
        cout << "The first " << n << " Fibonacci numbers: "   << "\n";
        for (int i=0; i < n; i++) {
        cout << ++j << " : " << fib(i) << "\n";
        }
        getch();
        return 0;
}
```

Compile and run it for n= 100 and modify it as explained in the comments.

1. What is the maximum Fibonacci number you can compute for the different integer
    definitions?

Compile and run the following program

```
/*
 * T R A C E.CPP
 *
 * C++ An useful trace makro.
 *
 *
 */
#include <iostream.h>
#include <conio.h>
#define TRACE cout<<__FILE__<< ", line: " <<\ // \ is the multiline macro operator
        __LINE__<<", function: "<< __FUNC__<<"\n"\
        <<"date: " <<__DATE__<<", time: " <<__TIME__<<endl;
```

```
void test()
        {
        int i = 10;
        }
int main()
{
        TRACE test;
        getch();
        return 0;
}
```

Exercise 13.2

1. Use one or more macros to define the Fibonacci function. How efficient is it?
2. Now sum up what you have learned about integers, the Fibonacci function and the use of macros.
3. What is meant by *"communis error facit jus"?*
4. How would you interpret that in the context of the C++ Language.
5. Do you know other (scripting) languages?
   - If yes, how error-prone are they?
   - If yes, how error-prone are they compared to C++?
   - Can these languages be compared to C++?
6. Look at the assembler statements for one or more macro expansions.

14. How to use the Help system more effectively.
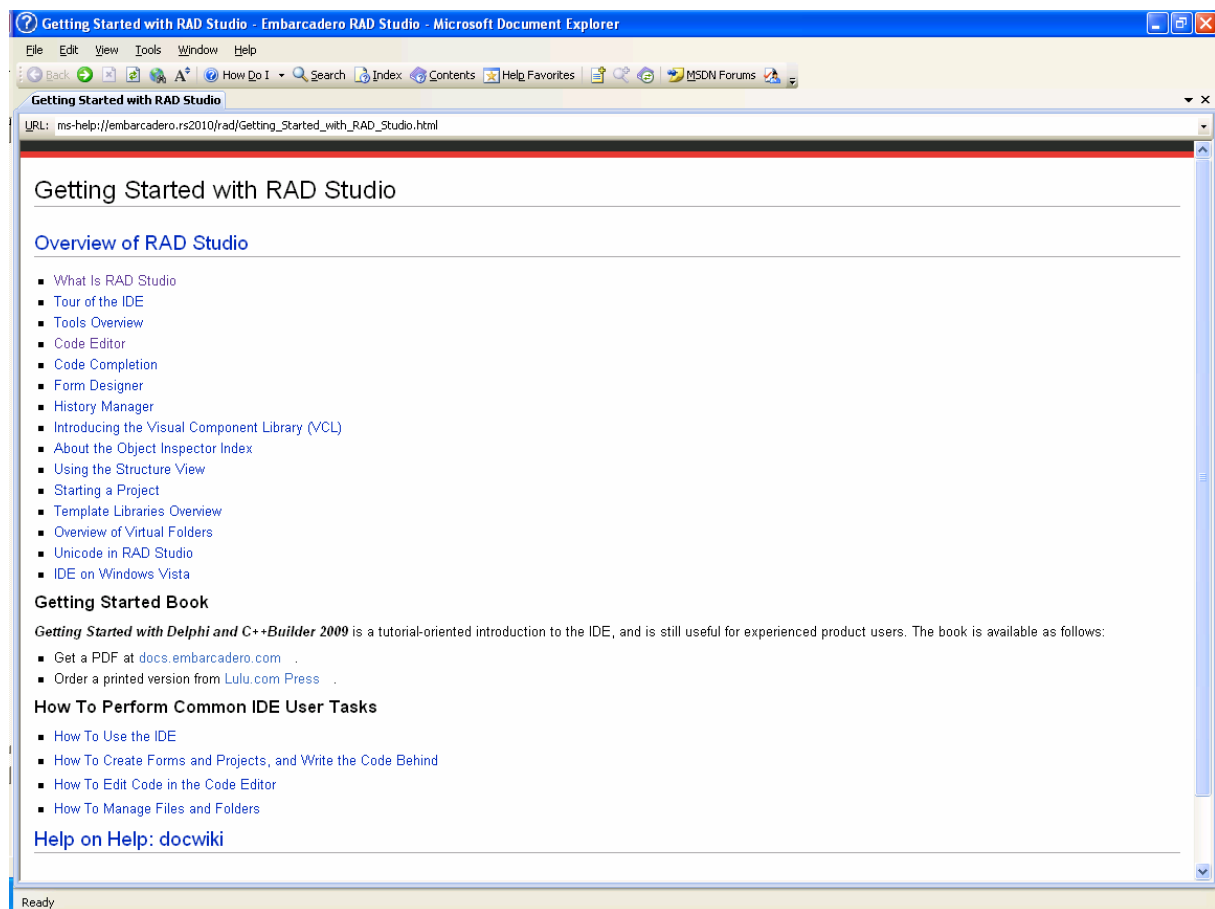
The screen dump on the other page shows the help system after we have clicked on | documentation | Getting Started | (your view may be different if "Sync with Table of Contents" is active – see below). From the subject line you will note that RAD Studio is based on "the Microsoft Document Explorer" that is a good system to organise the help content, index and search topics that you find interesting. Below the main menu, you see a customable toolbar, the small arrow to the right of the "MSDN Forums" and the "Report a Bug" icons. When you mouse over the arrow you see that you can set | Toolbar Options | Add or Remove Buttons | Standard | or | Customize … | The icon we are looking for is | Add to Help Favourites | , | Save Search | and | Sync with Table of Contents |. As can be seen from there toolbar icons, all three icons are already checked, the three icons to the left of the "MSDN Forums" icon. If you click | Add to Help Favourites | this is the new look. You will note that
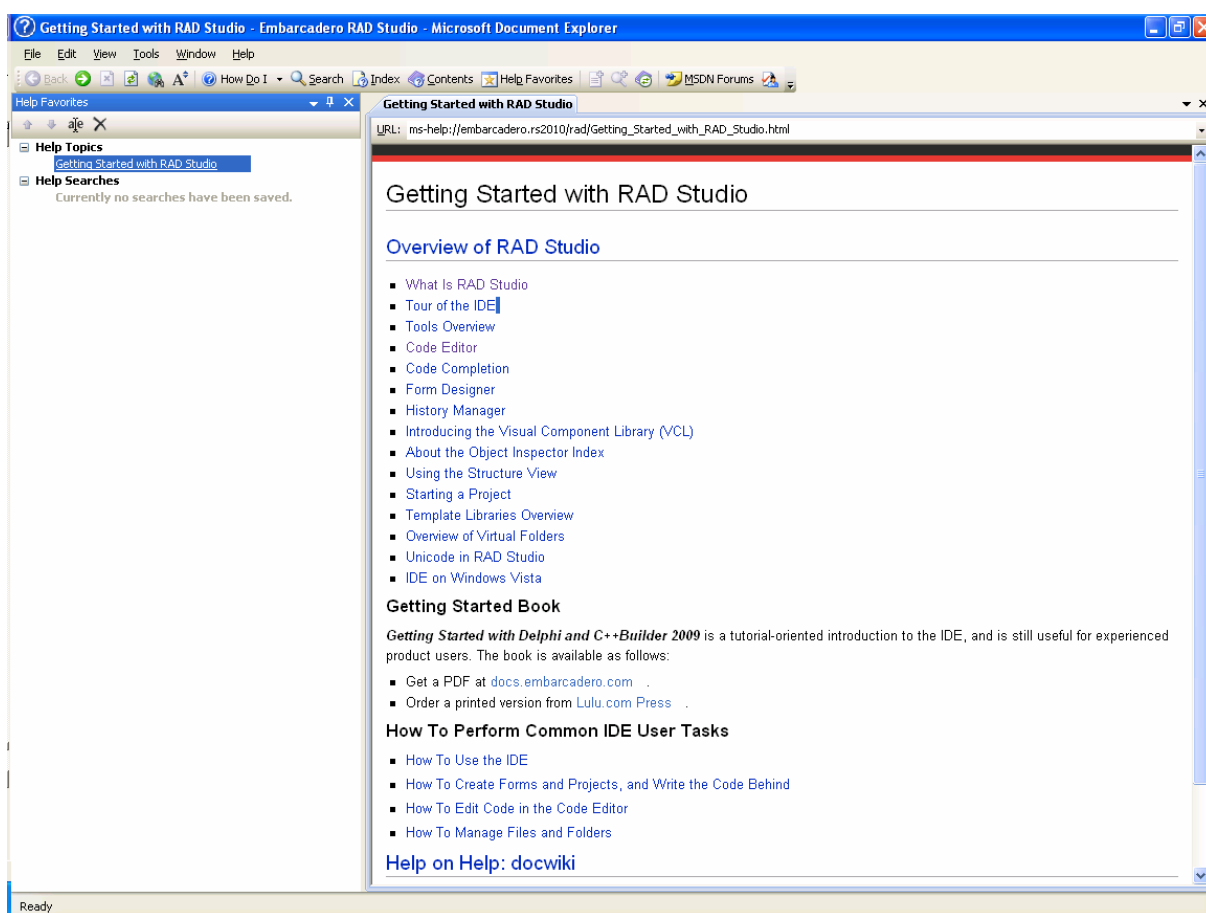
the important subject "Getting Started with RAD Studio" is now added to your "Help Favourites." For clarity, close the help system and click | Help | Table of Contents | Then Filter by:
| RAD Studio | Code Samples | C++ | and scroll down to MediaPlayer (see the screen dump on the next page). **Note that the usual T before the component as in TMediaPlayer is not used.**   If you want to add this code example to your Help Favourites, click the "Add to Help Favourites | icon. If you "Sync with Table of Contents", the second screen dump shows how it looks on my computer screen.   You can now see the information and code examples that are right at your finger tips and can be seamlessly organized by yourself. Not every component is documented, and to me it seems that the documentation has stopped with O. OnResize is the last documented subject in my version. The code example for the TMediaPlayer component may give you additional input to solve Exercise 7.1 in
http://www.oopschool.com/books/CPB2009.pdf
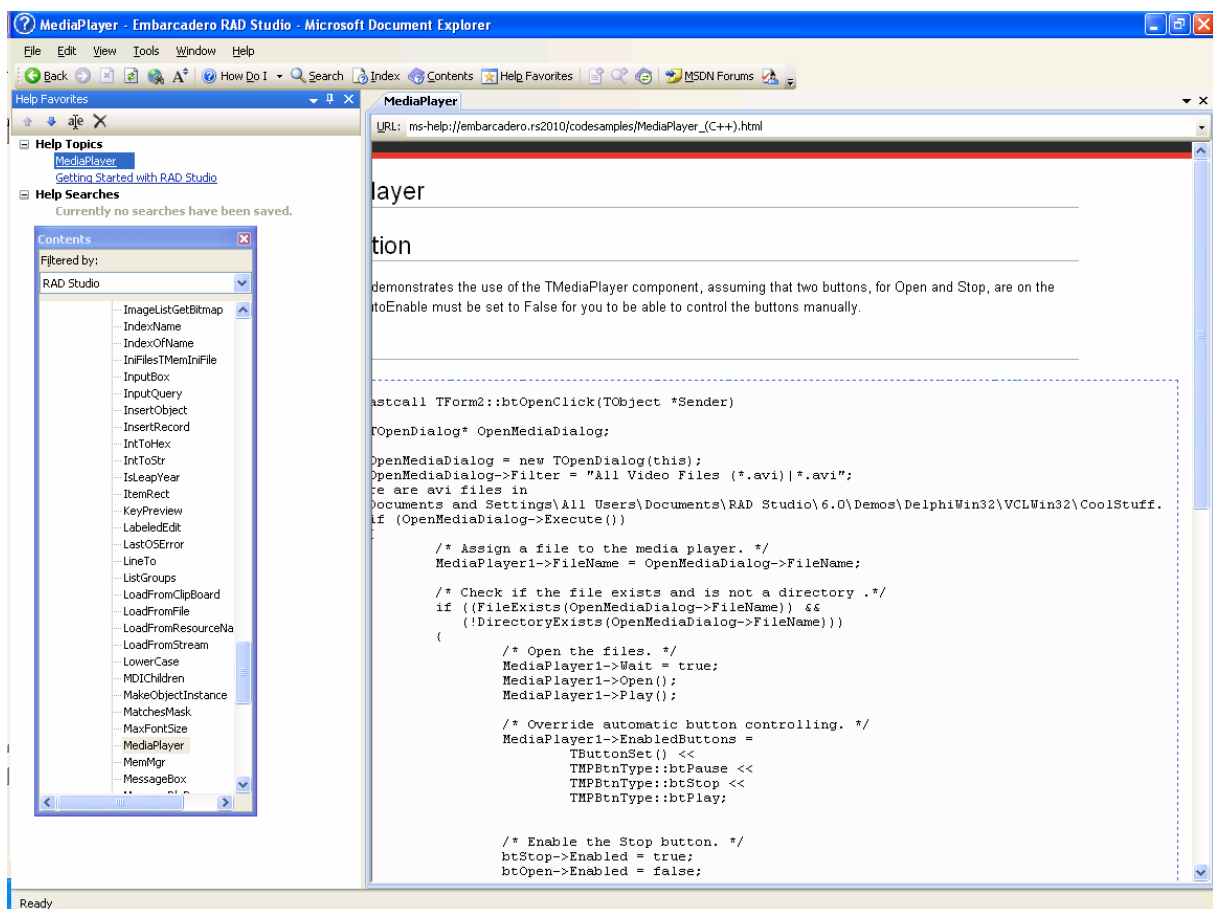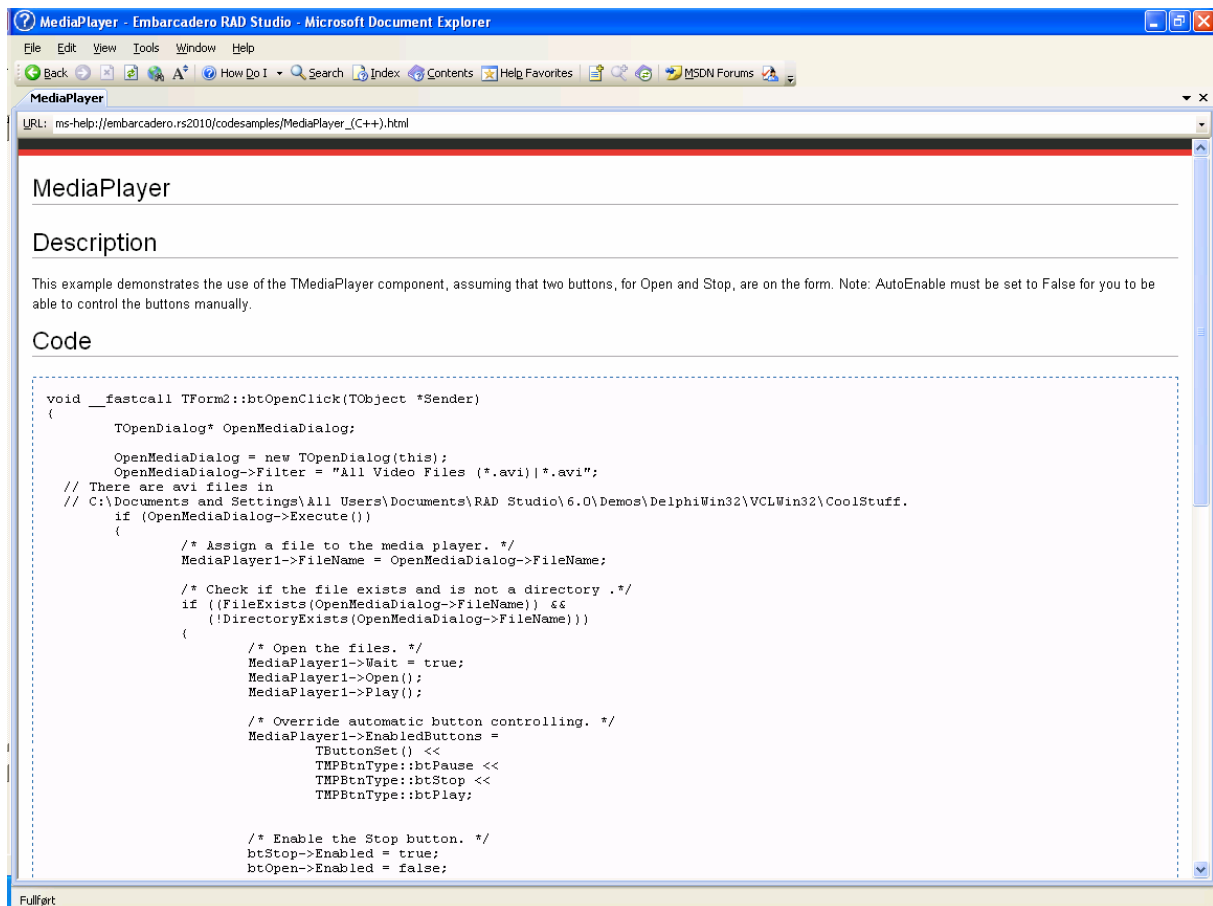
Exercise 14. 1.

Look up the subject in help system and build your own help favourites with the following
headings.

1. Dinkum C++ Library.

 What does this mean:
 "The Standard C++ library works in conjunction with the 18 headers from the
 Standard C library, *sometimes with small alterations. The headers come in two*
 forms, new and traditional. The new-form headers are"

2. C Library Overview.
3. Characters.
4. <algorithm>, binary search
 hint: ms-help://embarcadero.rs2010/Dinkumware/algorith/binary_search.html
5. <wchar.h>
6. <math.h> HUGE_VAL
 hint: ms-help://embarcadero.rs2010/Dinkumware/math/HUGE_VAL.html
7. Boost Libraries
8. C++ Library overview.
9. Unicode for C++ hint (see screen dump)
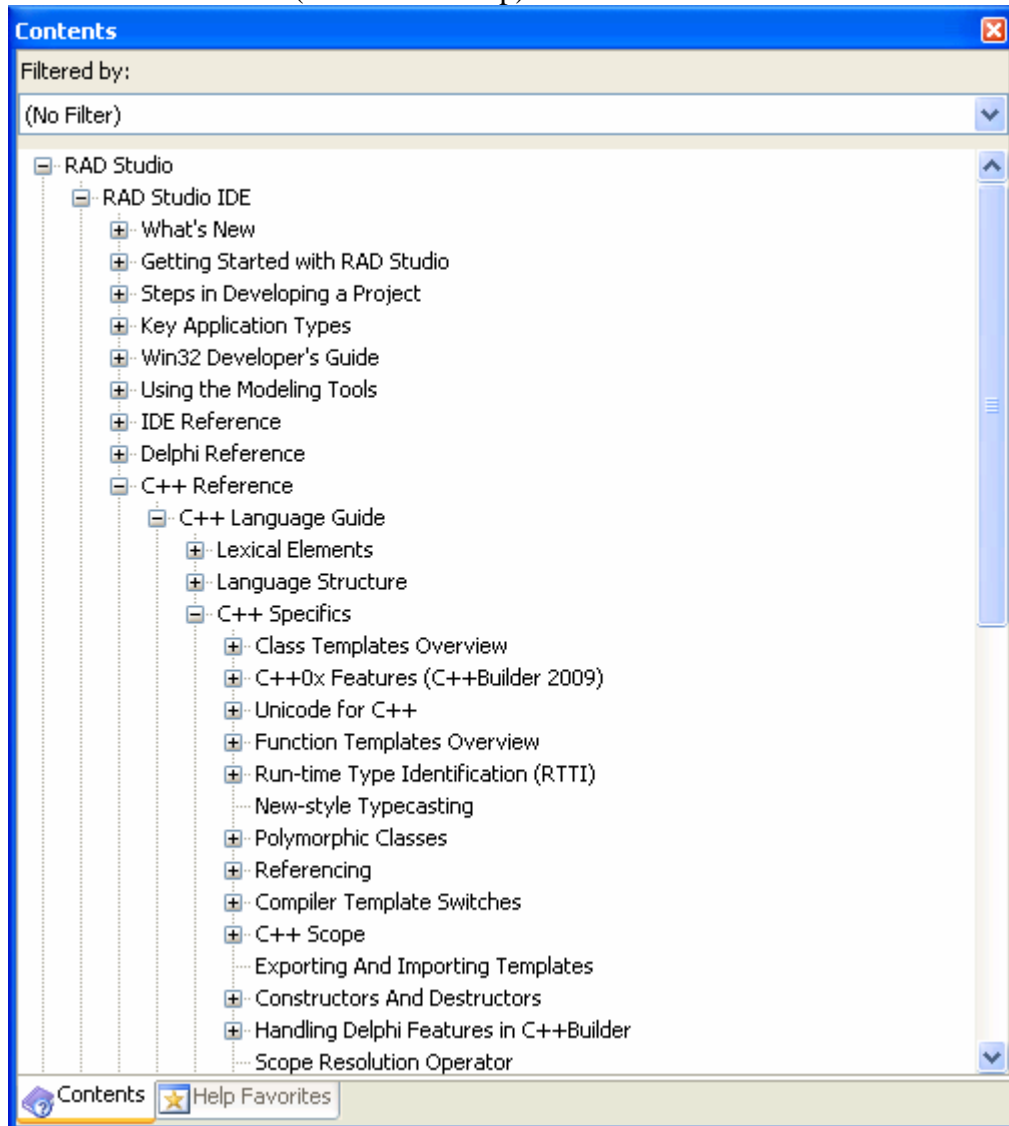
10. Handling Delphi Features in C++ Builder Index
11. The Preprocessor index. Hint screendump.
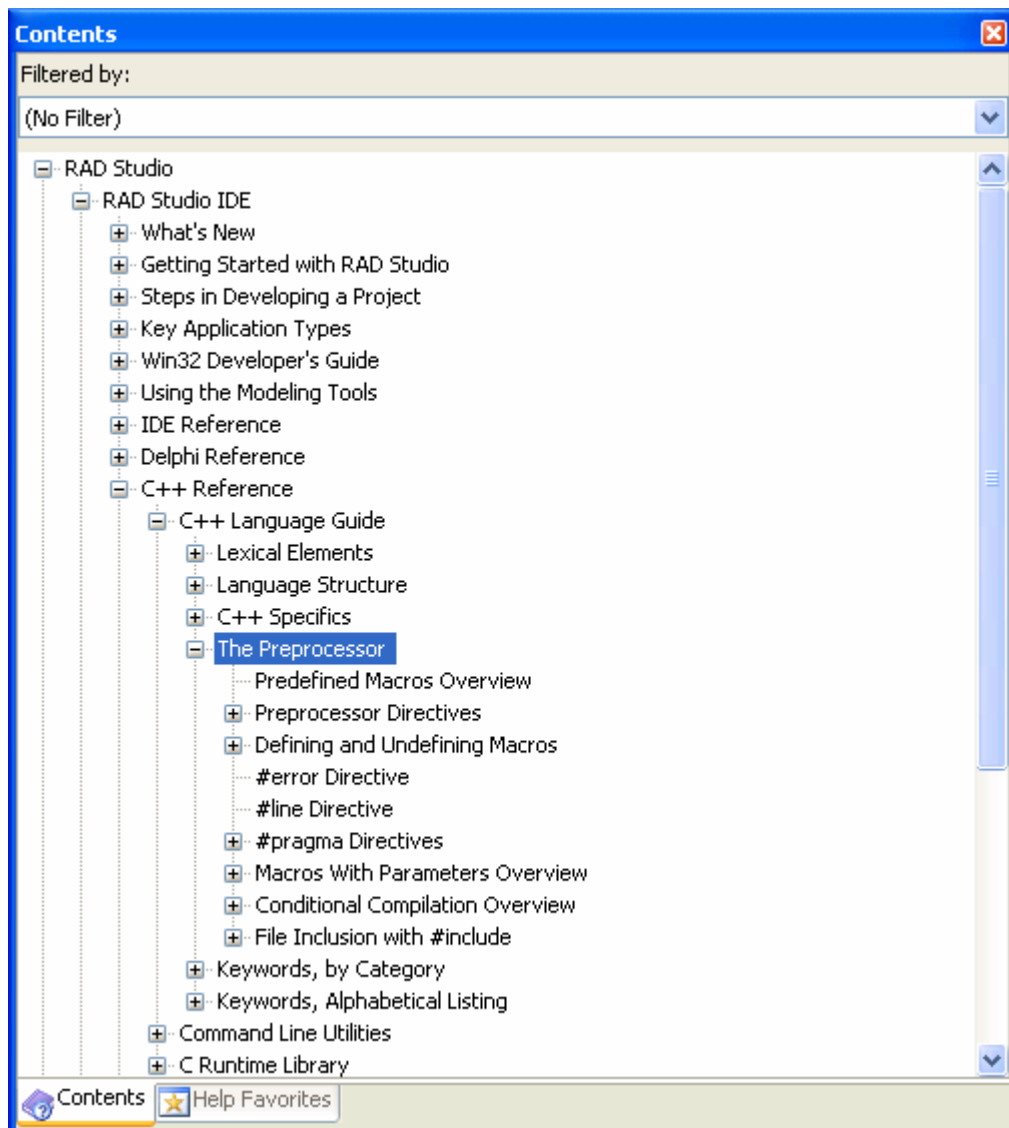
    Want is meant by conditional compilation?
12. The Fundamental Types.
13. C++ Error and Warning Messages. Hint: See above screen dump.
14. C++ Class Explorer Topics: Hint ??
15. Using Intraweb Components Hint (See screen dump on next page if you don't find it).
16. Class Explorer (C++).
    Hint (See second screen dump on next page if you don't find it). Then you have to look in one of the subcategories.
17. Predifined Macros (Easy if you have used your eyes of look for)
18. The fundamental Types.
19. Constructors And Destructors Index.
20. The main() Function index.
21. Steps in developing a Project.

Contents

Filtered by:

(No Filter)

- RAD Studio
  - RAD Studio IDE
    - What's New
    - Getting Started with RAD Studio
    - Steps in Developing a Project
    - Key Application Types
    - Win32 Developer's Guide
      - Programming with RAD Studio
      - Developing Database Applications
      - Developing DataSnap Applications
      - Developing Internet Applications
        - Creating Internet Server Applications
        - Using Web Broker
        - Using WebSnap
        - Using IntraWeb (VCL for the Web)
        - Working with XML documents
        - Using Web Services
        - Working with Sockets
      - Developing COM-Based Applications
      - Component Writer's Guide
    - Using the Modeling Tools
    - IDE Reference
    - Delphi Reference
    - C++ Reference
    - Subject Index
  - VCL Reference
  - Code Samples
- Dinkumware
- Internet Direct (Indy) 10

Contents | Help Favorites

Contents

Filtered by:

(No Filter)

- Error Messages
- File Menu
- Help Menu
- HTML Elements
- Insert Menu
- Modeling Dialog Boxes
- Project Menu
- Property Editors
- Run Menu
- Refactor Menu
- Search menu
- Testing Wizards
- Tools Menu
- View Menu
- Win View
  - API Categories
  - Keyboard Mappings
  - Command Line Switches and File Extensions
- Delphi Reference
- C++ Reference
- Subject Index
  - VCL Reference
  - Code Samples
- Dinkumware
- Internet Direct (Indy) 10
- Rave Reports
- TeeChart 8 Standard
- Microsoft Windows Platform SDK

Contents | Help Favorites

15. How to build a simple IntraWeb VCL Web Application.

I started here: ms-help://embarcadero.rs2010/rad/Getting_Started_with_IntraWeb.html

| File | New | Other | VCL for the Web | VCL for the Web Application Wi… | OK | OK |

That means that I choose a

| StandAloneApplication | Create User Session | HTML 4.0 |

As explained here:

ms-help://embarcadero.rs2010/rad/Editing_the_Main_Form.html

I edited the main form Unit1.cpp using the IW Standard Tool Palette:

Here
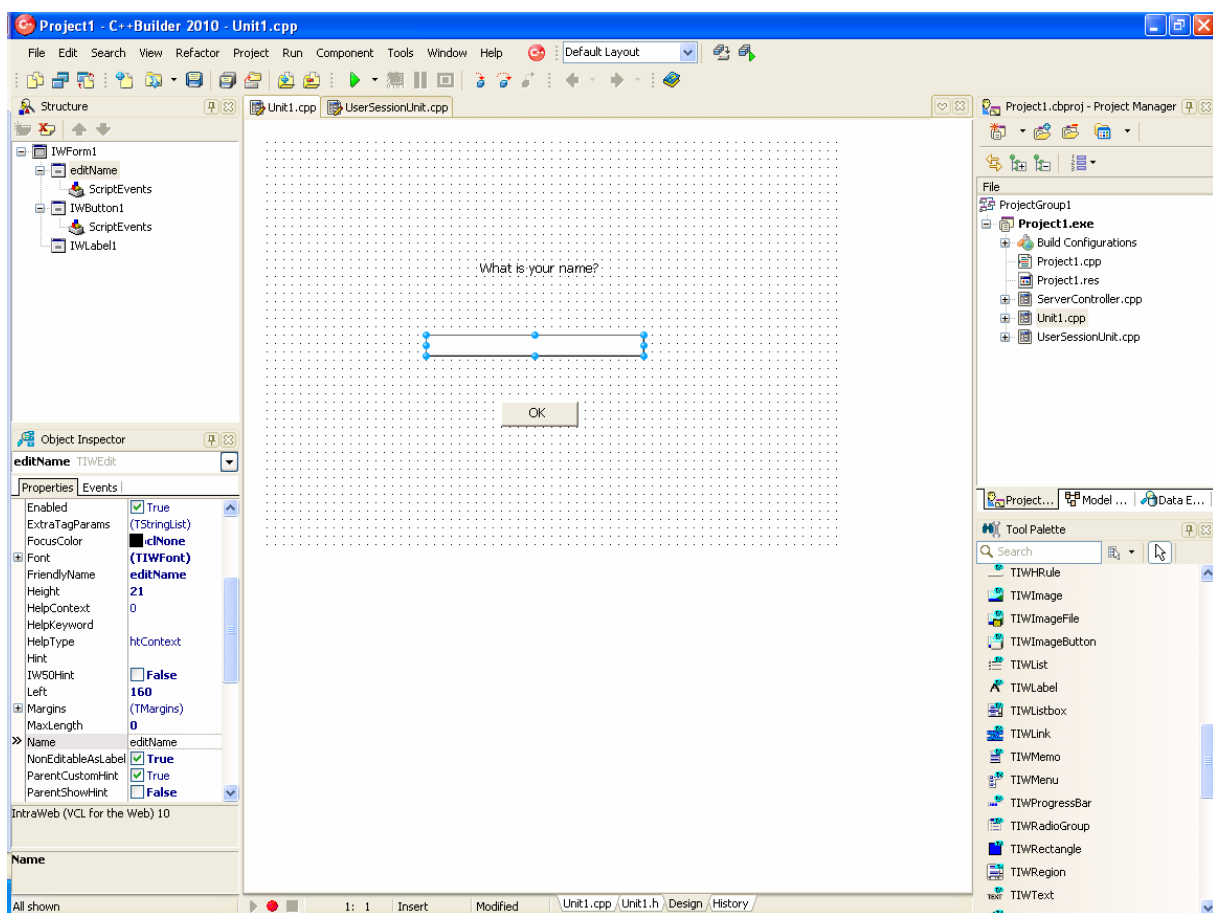
is the form so long.

Double clicking on the OK button takes me to the eventhandler
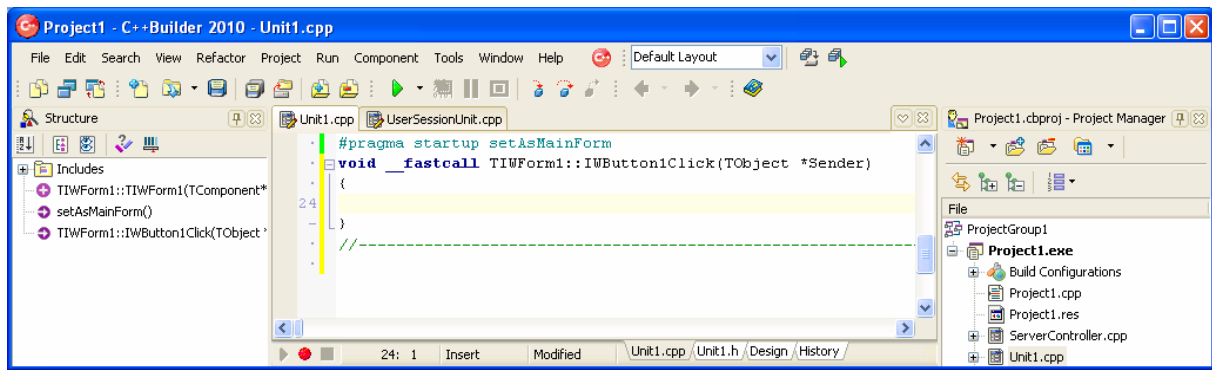
Where I write in the code:

```
void __fastcall TIWForm1::IWButton1Click(TObject *Sender)
{

    UnicodeString s;
    s=editName->Text;
    if (s.Length() ==0) {
    WebApplication->ShowMessage("Please enter your name.");
    }
    else {
    WebApplication->ShowMessage("Hello, " + s +"!");
    editName->Text = "";
    }
}
```

that gives an error

[BCC32 Error] ServerController.cpp(27): E2227 Extra parameter in call to _fastcall TIWServerControllerBase::SetServerControllerClass()

on the following line

```
 TIWServerController::SetServerControllerClass(__classid(TIWServerController));
```

That according to the second post of this

ms-help://embarcadero.rs2010/rad/Getting_Started_with_IntraWeb.html

thread has to be changed to

```
Controller::SetServerControllerClass();
```

Of the ServerController.cpp file.

Hitting F9 now, gives this error

[BCC32 Error] Unit1.cpp(26): E2227 Extra parameter in call to _fastcall TIWAppForm::SetAsMainForm()

on line 26

```
void setAsMainForm() {
  TIWForm1::SetAsMainForm(__classid(TIWForm1));
}
```

of the main form file, Unit1.cpp that again has to be changed to.

```
void setAsMainForm() {
  TIWForm1::SetAsMainForm();
}
```

and now the project compiles and links correctly indicated by this
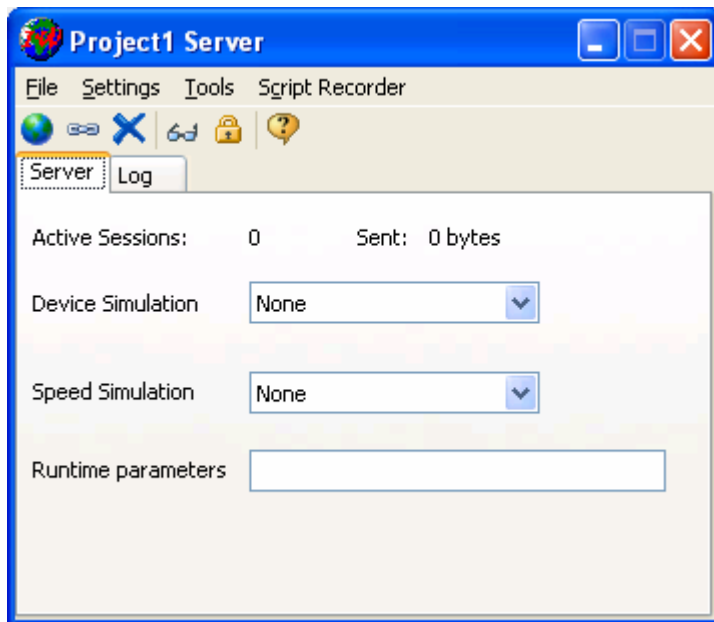
screen dump. Now: | File | Execute | on the above main menu gives
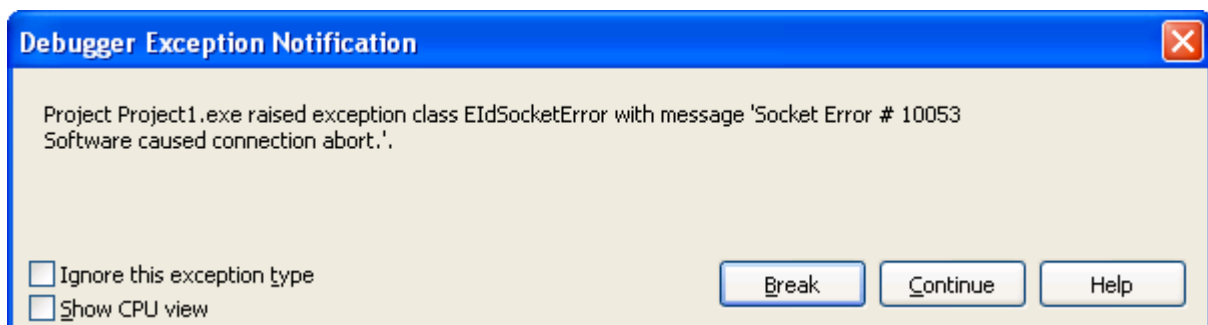
where I hit continue. You see the final result on the next screen dump.
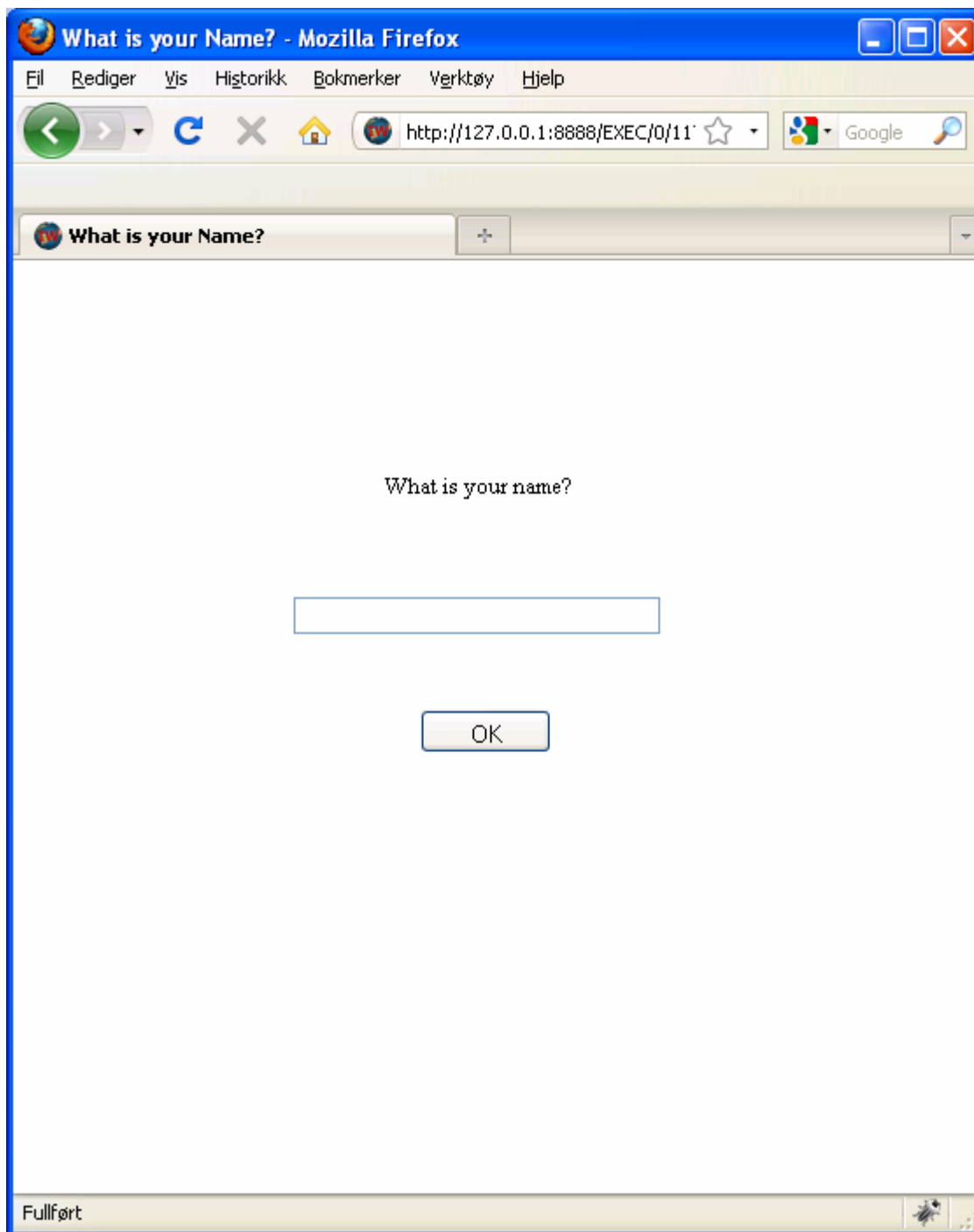
is the application with this result after writing in my name.
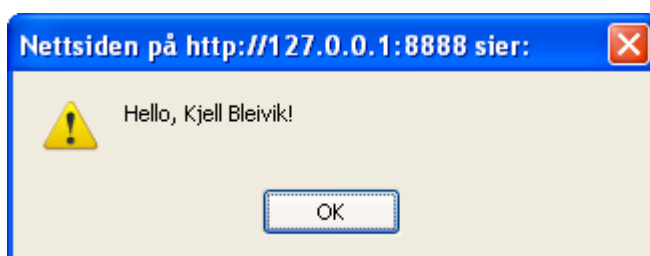
16. Releasing your application with your own logo and version information.

Let us make a simple application with an edit file and two buttons. When you hit the left button, "Yellow", the edit field is set to yellow. The other button, "default" set the edit field back to the default white colour. You should be able to do this yourself now. If not look at the event handlers for the two buttons at the end of this chapter. Make a folder with the name Release and a sub folder with the name temp. Save the cpp unit file by the name ReleaseU.cpp and the project name with the name Release in the release folder.

1. Compile and run the program.

Here

is a screen dump of the application after I have hit the Yellow button. Now locate the Release folder on your computer and look at the code. As you will see, from the first screen dump on the next page, the Debug sub folder with the application is 37 MB large. From the next screen dump you will note that the Release application is 38.5 kb.

2. Configure the release environment | Project | Configuration Manager… | like

the fourth screen dump below.
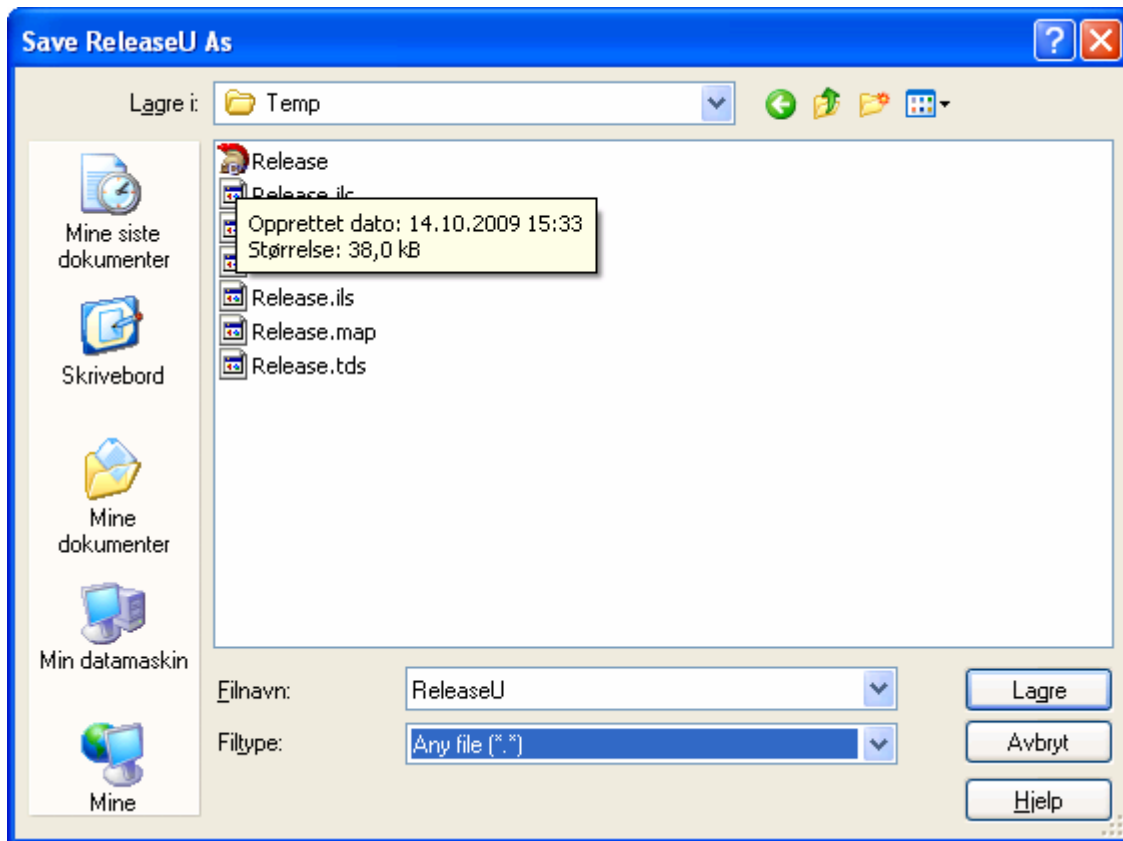
3. | Project | Options |

In the "Build Configuration" scroll field choose "Release" and set the "Final output" to the "Temp" sub folder. See the fifth screen dump below. Hit OK when the configuration is OK. If you want you can choose your own application Icon on | Project | Application | as seen from the sixth screen dump. You will now note that there are two subfolders of the Release project folder, Temp and Release respectively. The final application that is saved in the "Temp" subfolder is marginally smaller than the one generated with the default "Base" "Build Configuration". For larger applications the difference can be larger. You will also see that the number of files in the "Temp" folder is less than in the standard "Debug" subfolder of the project.

Project Options for Release.exe - Release Configuration

- Directories and Conditionals
- C++ Compiler
  - Advanced
  - Compatibility
  - Debugging
  - Directories and Conditionals
  - Optimizations
  - Output
  - Pre-compiled headers
  - Warnings
- C++ Linker
  - Output
  - Warnings
- Delphi Compiler
  - Compiling
  - Hints and Warnings
  - Linking
- Resource Compiler
  - Directories and Conditionals
- Turbo Assembler

Build Configuration: Release

BPI/LIB output	
Conditional defines	**NDEBUG**
Final output directory	**Temp**
Include path	C:\CppProjects\Release\;$(CG_BO
Intermediate output	**Release**
Library path	**$(BDS)\lib\release;C:\CppProj**
_TCHAR maps to	char

☐ Default   OK   Cancel   Help

Project Options for Release.exe - Release Configuration

- Warnings
- Delphi Compiler
  - Compiling
  - Hints and Warnings
  - Linking
- Resource Compiler
  - Directories and Conditionals
- Turbo Assembler
  - Directories and Conditionals
  - Warnings
- Build Events
- Build Order
- Application
- Debugger
  - Symbol Tables
  - Environment Block
- Forms
- Packages
- Version Info
- Project Properties

Application settings

Title: [   ]

Help file: [   ]   Browse...

Icon: [icon]   Load Icon...

☑ Enable runtime themes

Output settings

Target file extension: [exe]

☐ Default   OK   Cancel   Help

Here is the event handlers for the two buttons:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Edit1->Color = clYellow;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Edit1->Color = clWindow;
}
```

Exercise 16.1

1. Go back to exercise 5.1.
2. Build the calculator with your own application icon.
3. On the | Projects | Version Info | menu, set the version info and build a final release version.
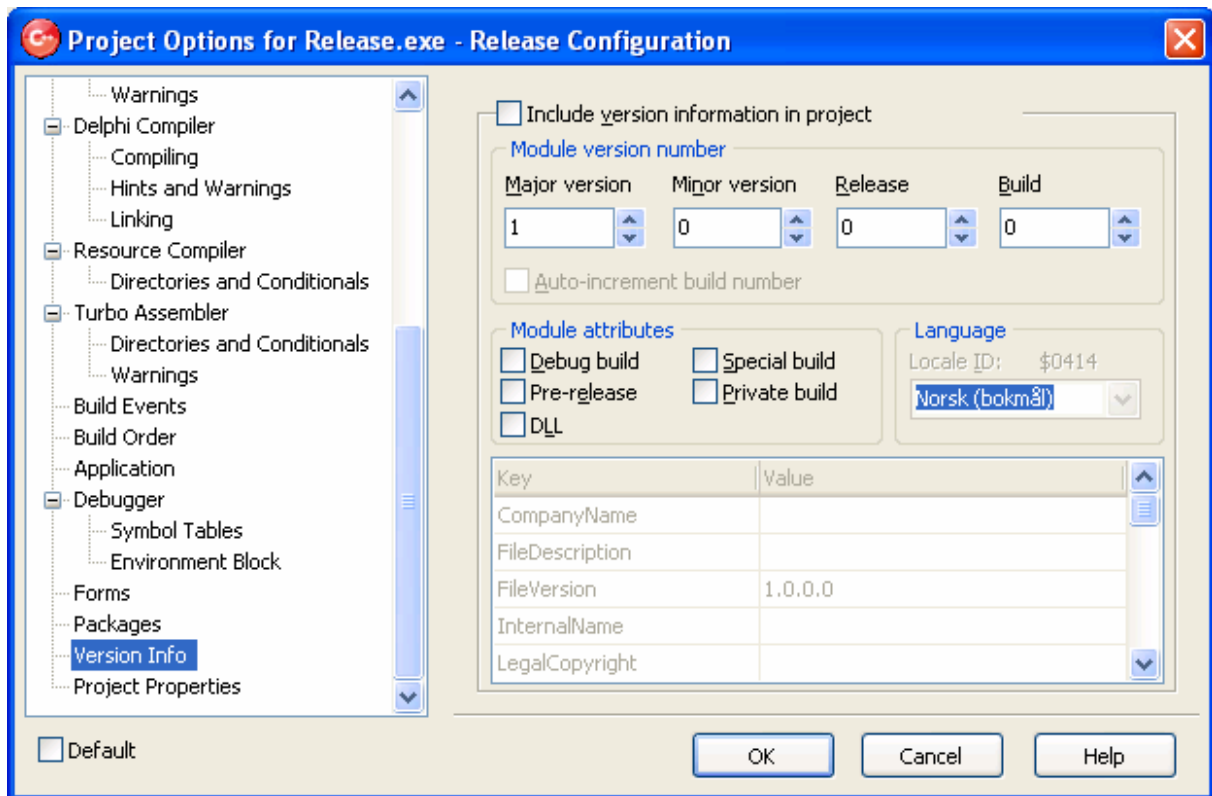
17. Some differences between C++Builder 2009 Professional and C++Builder 2010 Professional.

We shall not dive deep into this subject, since it is well documented via the Welcome page:
| Documentation | Rad Studio Help |

| What's New in Delphi and C++Builder 2010 |
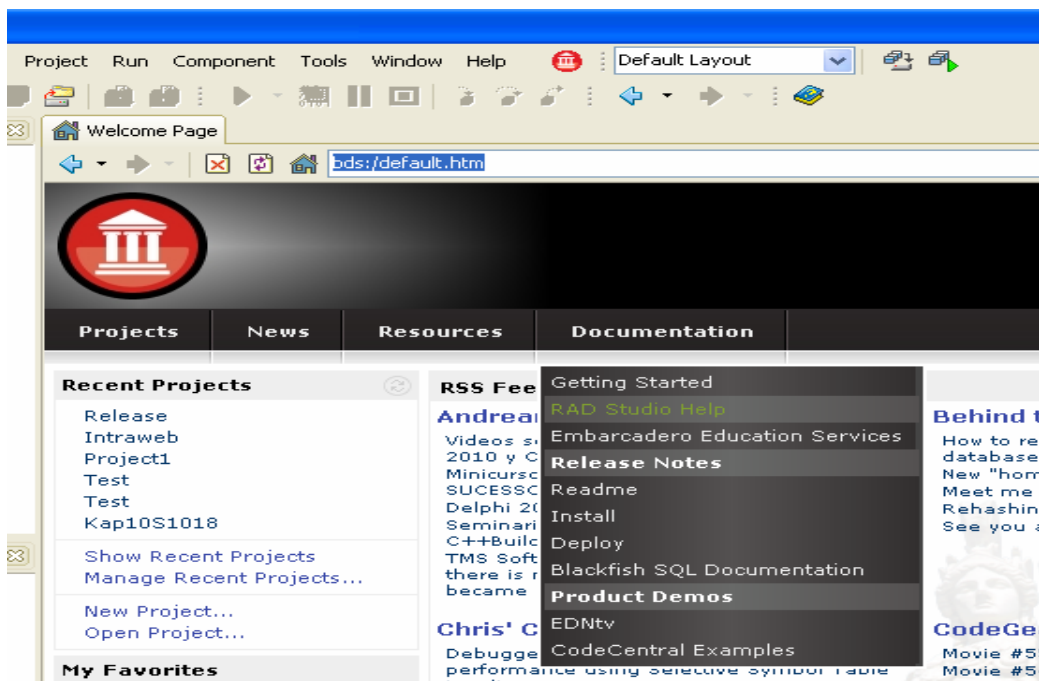
The Class explorer is one big difference. See Exercise 14.1 how you find information on the Class Explorer. In this short chapter, we will concentrate on the Help system itself, since our

experience is that being able to look up subjects yourself is often the most important thing you learn when you start to use a new platform. Simply, learn how to use the manuals and / or help systems. So start by locating the above page, and scroll down to the heading "Online Help Changes." When you have read that short introduction, you click the link with anchor text "What's New in the Help, and Where Is Everything". On that page, scroll down to "Where is Everything in the Help" to get an overview. Let us try an experiment:

In the Help index look for: TEdit. Here
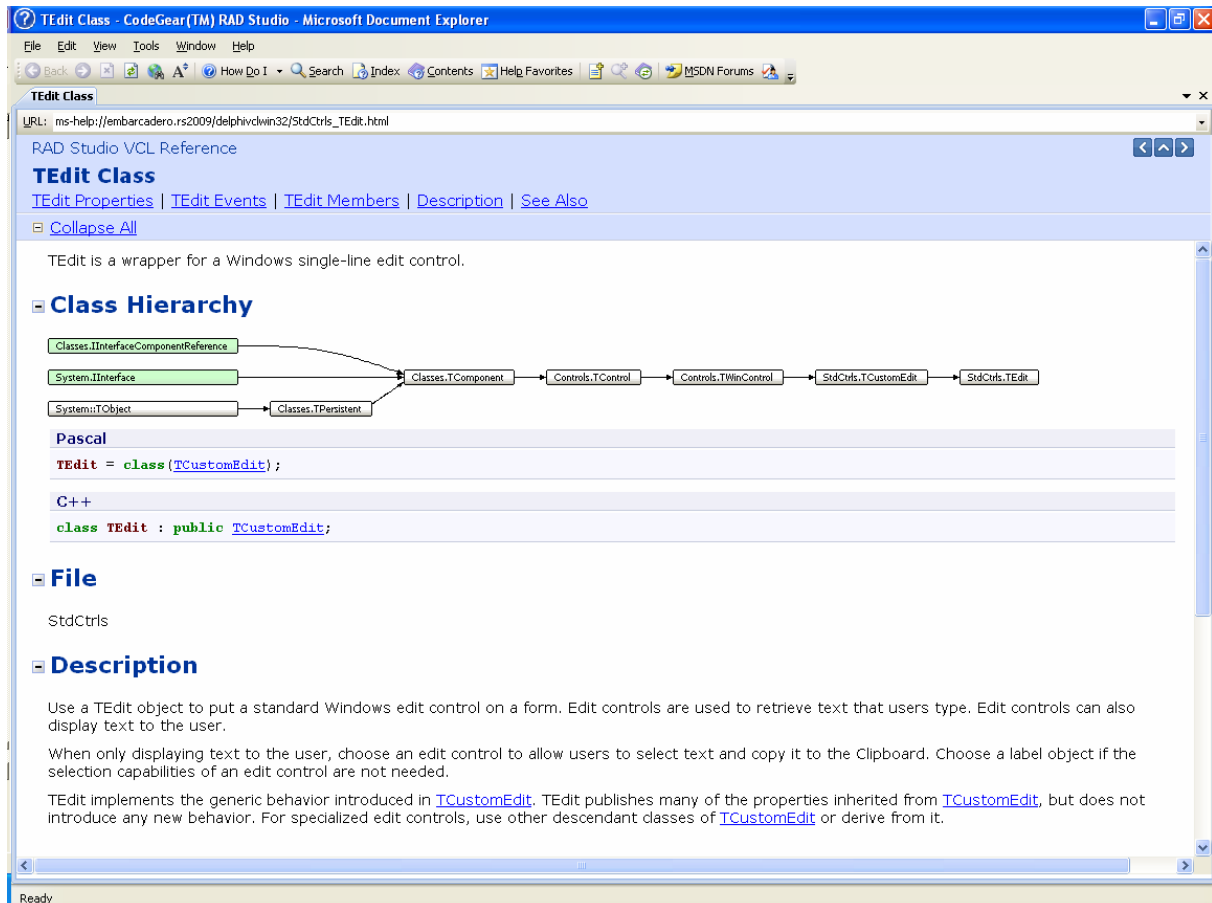
is a screen dump from C++Builder 2009 that should be familiar to you if you use that version. On the next page you see a screen dump from the 2010 version of the C++Builder. The graphical description starting with the TEdit component's place in the Class Hiearchy is no longer the default page in the new help system. And the component's elements, that is the component's properties, events and methods are no longer seen at the top of the page.   The organization of the subject is very different. Which is better is a matter of taste. Now let us continue our journey in the help system. In both versions start a new VCL forms Application C++ Builder project and locate the "Internet" tool palette. In both versions click the TCppWebBrowser component and then the TWebBrowser component and hit F1. You should get the same message "No help found for TCppWebBrowser" in both versions of the C++ Builder. But when you click F1 when TWebBrowser is selected on the tool palette in the 2009 version, you get a similar Class Hiearchy help page as for the TEdit component. In the 2010 version it is different, see the second screen dump on the next page.   There are two choices, but with a fast look, I can not see a difference between the two help pages that is loaded, but as for the TEdit component the help documentation is different.

Exercise 17.1.

In both versions of the C++Builder, put a TWebBrowser component on the form and:

1. Click the component and then F1.
2. In the Object Inspector chose the Align property and hit F1.
3. Then Chose the OnDownloadComplete Event and hit F1.
4. Double click on this event so you are taken to the following event handler

   void __fastcall TForm1::WebBrowser1DownloadComplete(TObject *Sender)
   {
   }

   Now mouse over the red text and hit F1 in both versions.
5. Note the differences and try to explain.

6. In version 2010: | View | C++ Class explorer | (or CTRL +-) and expand the VCL folder by clicking the + sign to the left and scroll down to TWebBrowser and expand it. If you hide the other windows, you should have a screen dump similar to this:
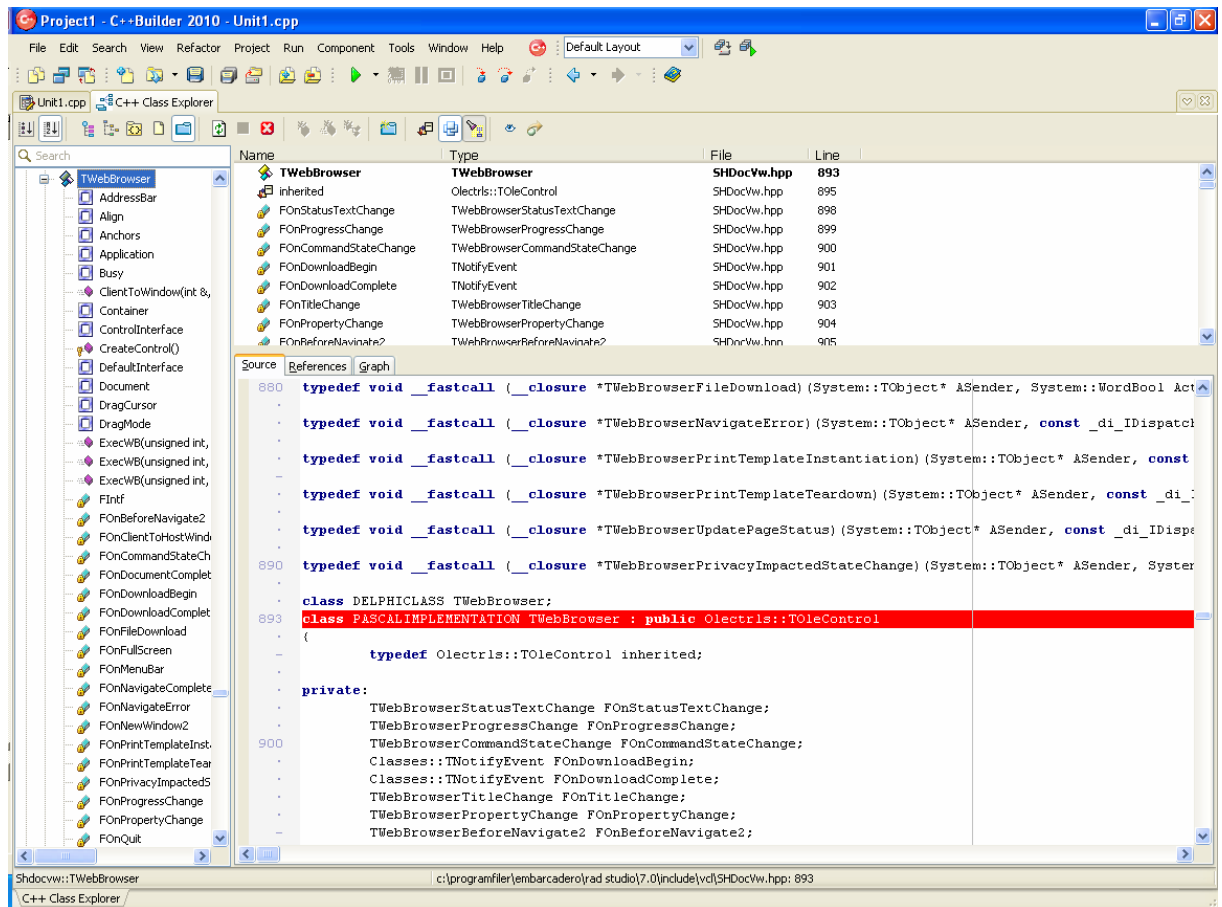
In the lower right window, you see three tabs, Source (the above view), References and Graph. Click the two folders and observe the content. Then Click on FOnDownloadBegin in the above window and the three tabs again. The screen dump on the next page shows my screen: Then double click on FOnDownloadBegin. The source code for the SHDocVw.hpp header file is shown where the following lines

Classes::TNotifyEvent FOnDownloadBegin;

class DELPHICLASS TWebBrowser;
class PASCALIMPLEMENTATION TWebBrowser : public Olectrls::TOleControl
{
        typedef Olectrls::TOleControl inherited;

private:
        TWebBrowserStatusTextChange FOnStatusTextChange;
        TWebBrowserProgressChange FOnProgressChange;
        TWebBrowserCommandStateChange FOnCommandStateChange;
        Classes::TNotifyEvent FOnDownloadBegin;

shows that TWebBrowser is a Delphi class and FOnDownloadBegin a private class, that implies that it is not available to other components. (A component is a class, but a class is not necessarily a component it the C++Builder IDE).

:

7. Comment on the advantages with the new Class Explorer, the online help wiki

    http://docwiki.embarcadero.com/RADStudio/en/Main_Page

    and the help system in earlier versions of the C++ Builder Professional.
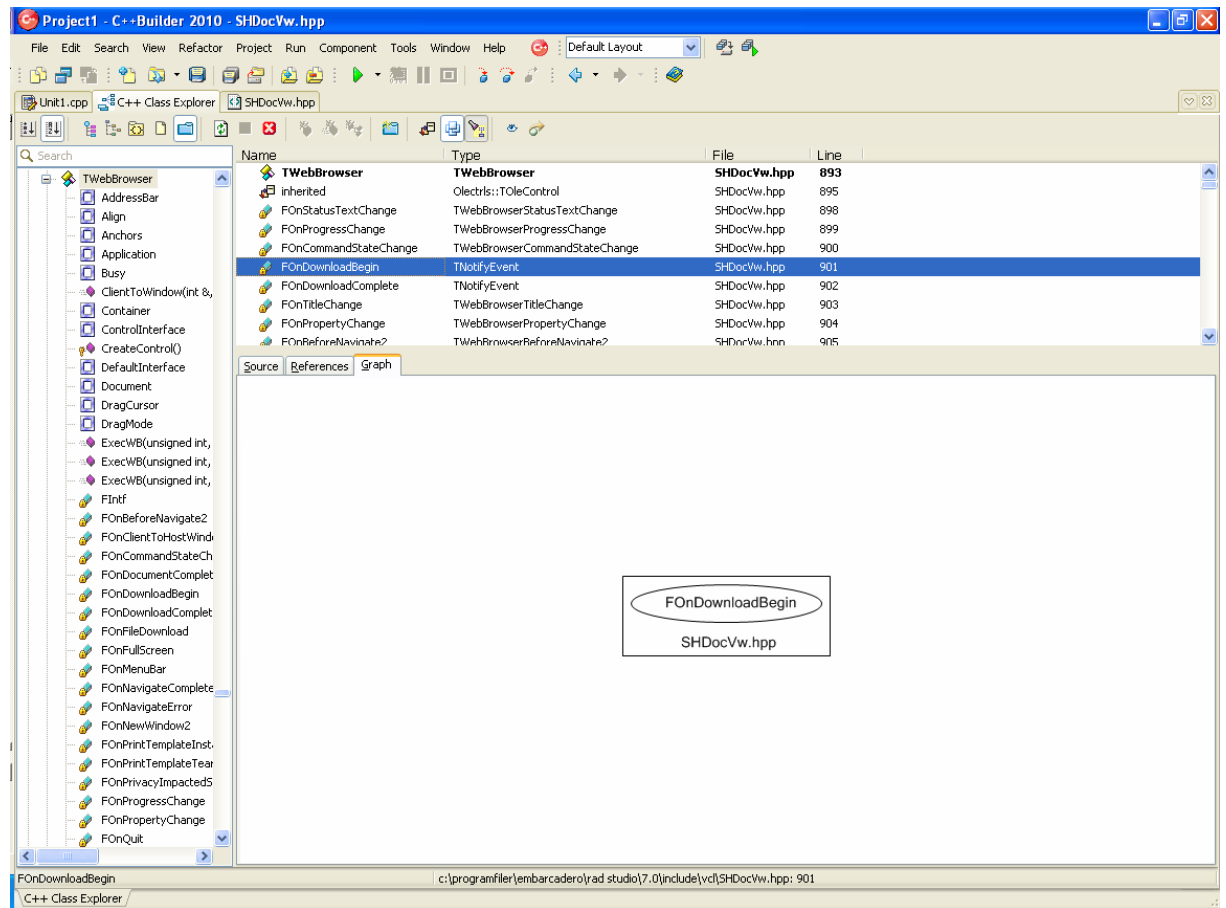
8. Produce the two screen dumps you see on the next page in the Class Explorer.
9. In the abs(int) function, look up the inline definition and go to the top of the header file and look at the explanation of stdlib.h.
10. How many subjects in the | Class explorer | RTL | starts with FtP, Search and URL respectively
11. Finally look up the function definition for ULongLongMult( ) with the paranthesis marked blue and hit F1.

Now, you should have a fairly good overview of the difference between the help system in C++Builder 2009 and C++Builder 2010. You should also know the potential in the C++ Class Explorer and how to organise your own help favourites. We have only skimmed the surface of the new IDE. We can not teach you how to program in this short document, but we hope that this journey has given you a little taste of the potential in this excellent development platform. You should now be able to locate C, C++ and assembler code on the internet that you can integrate into your own projects.
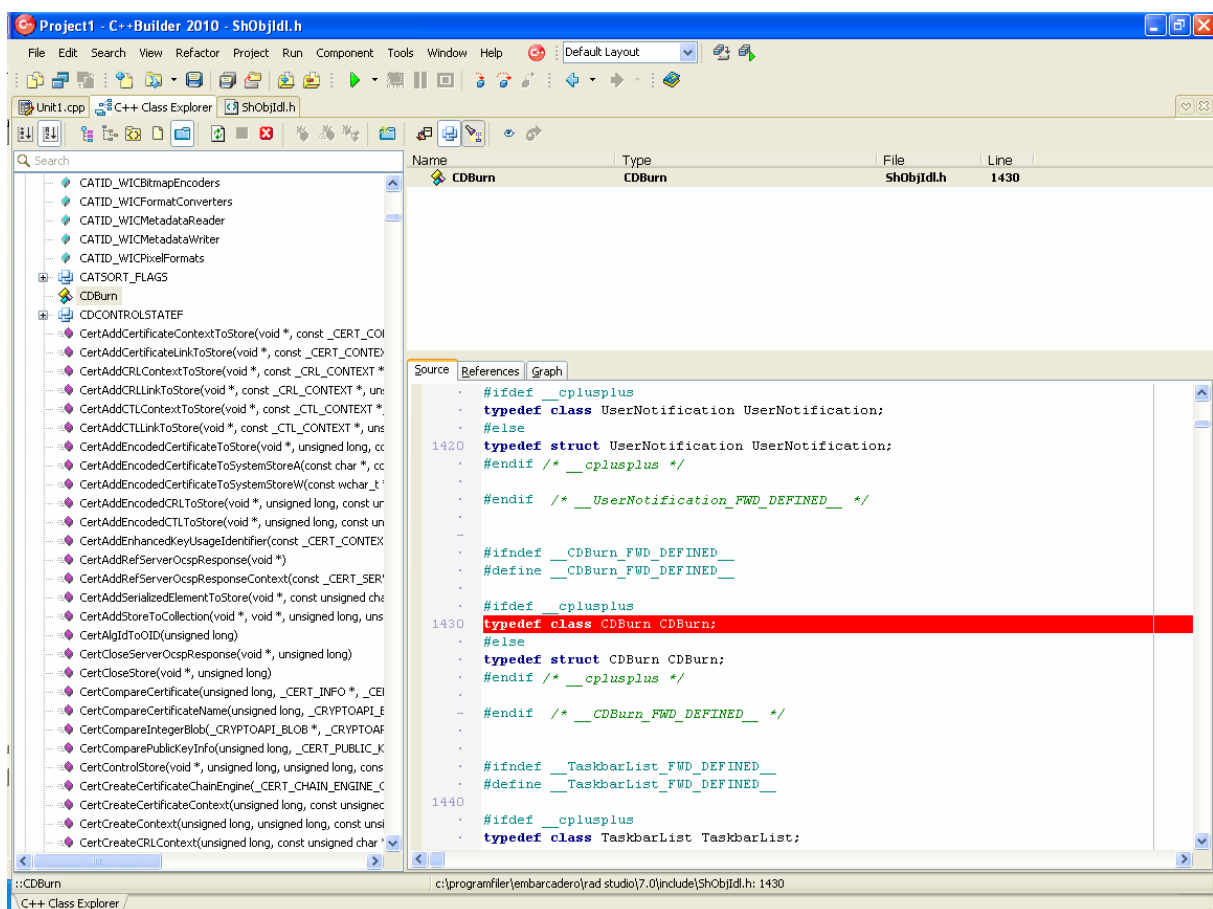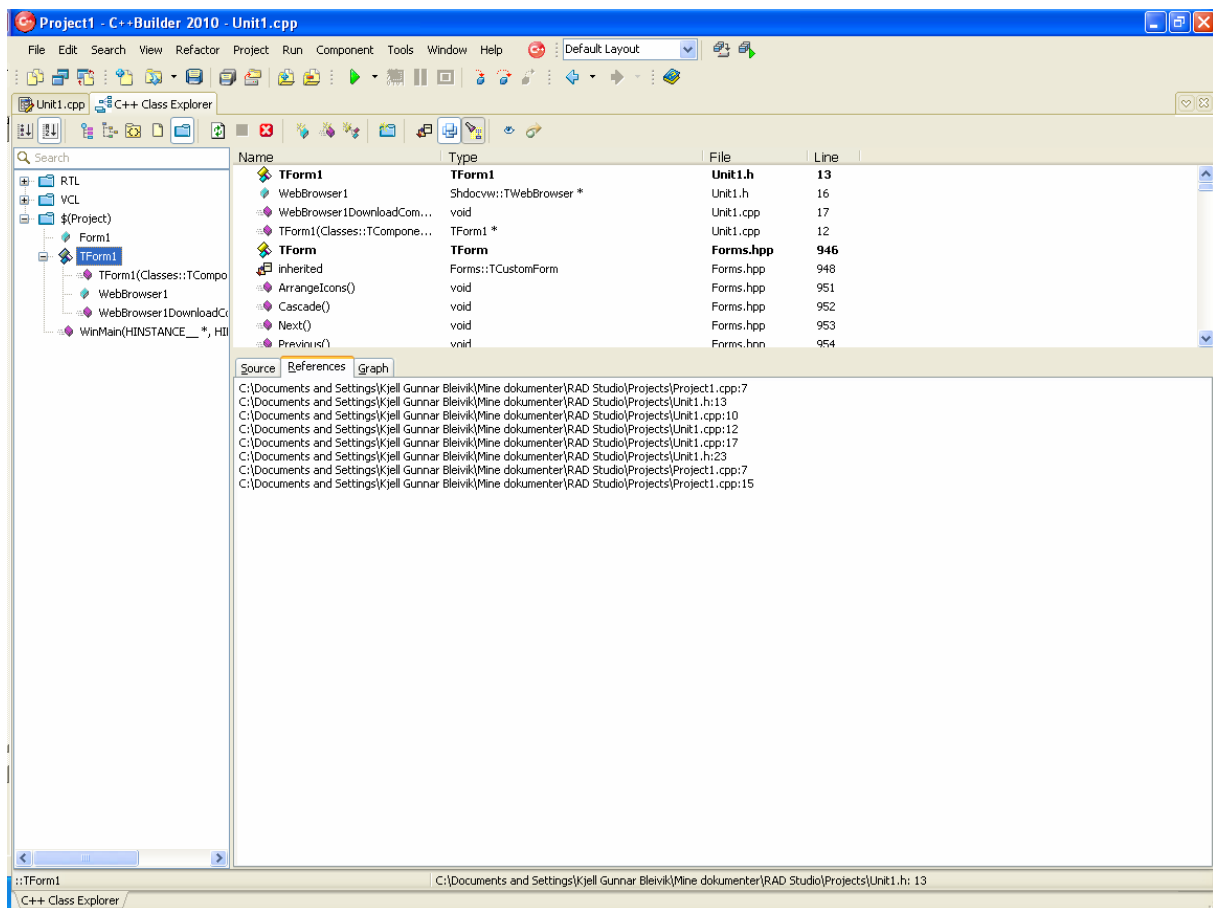
18. The desktop is on the web.

Idea: I want to port a simple calculator to the web. So let us start

| File | New | Other | VCL for the Web Application Wi… | Without User Session.
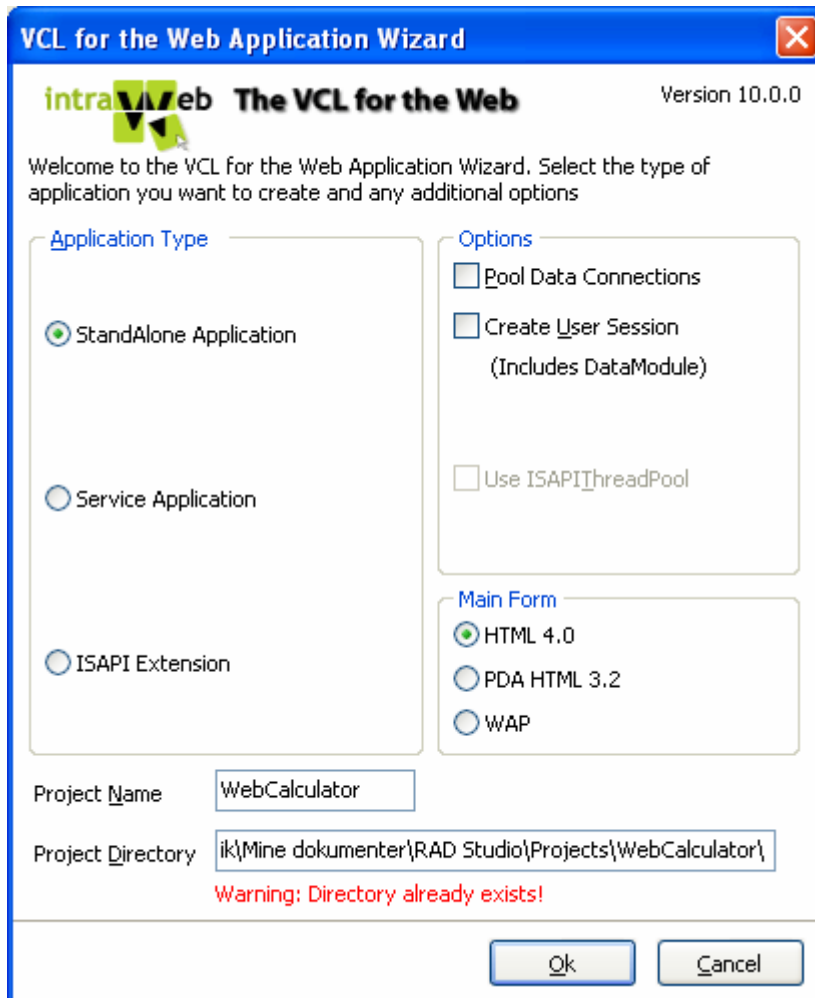
Note the Warning in red. I get it because I created that project directory before the project. You don't need to create it. The folder is automatically created for you with the same name as the Project Name. Then click OK. Save the project with the name WebCalculator and the form unit file with the name WebCalculatorU.cpp.

Compile and run the application to see that it functions so long. If you use C++Builder 2009 Professional, you should not get an error. As explained in chapter 15. "How to build a simple IntraWeb VCL Web Application" you get the same errors that is described there that have to be fixed in the same way. **So if you have both C++Builder 2009 Professional and C++Builder 2010 professional, you may prefer to use the first.** I used C++Builder 2010, so I can not guarantee that the rest function identically as explained below for the C++Builder 2009 version. If your application is blocked by your firewall, you have to allow it access. You may also get one or more debugging messages where you have to click continue every time. Finally you should have a blank web page with an URL starting like:

 http://127.0.0.1:8888/EXEC ....

or similar. If you view the source, you should see traditional HTML markup with embedded JavaScript code.

Now, close the program and continue to build the application.

To simplify our work, we need three toolbars, Position, Spacing and Align that is easiest activated by right clicking on the grey toolbar field below the blue status line and check the three boxes. On the Tool Palette choose the IW Standard Toolbar and put three TIWEdit components and 6 TIWButtons (it is fastest to double click on the TIWButton icon 6 times) on the WebCalculatorU.cpp form. You select a group of components, like the five buttons in the middle by dragging the mouse around them as shown in the screen dump. Note "5 items
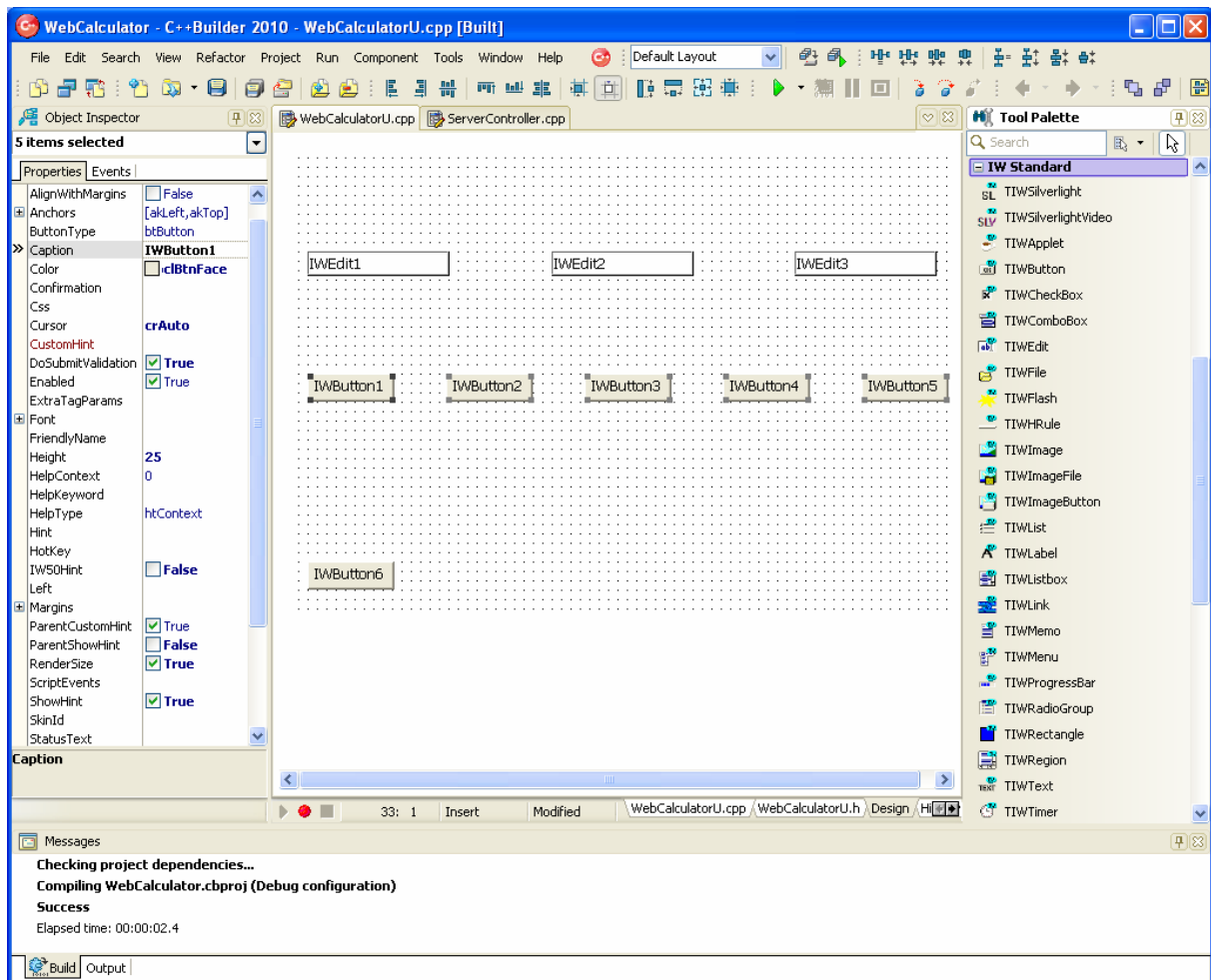
selected" in the upper left corner (The structure window is closed, as is the project manager window). Then it is easy to position, align and space the components equally by using the correct icons on the three design toolbars. Now it is time to write the event handlers. For IWButton1, write the following method:

IWEdit3->Text=(IntToStr(StrToInt(IWEdit1->Text) + StrToInt(IWEdit2->Text)));

and similar for the other buttons, with -, *, / and % operators. E.g. For IWButton5

IWEdit3->Text=(IntToStr(StrToInt(IWEdit1->Text) % StrToInt(IWEdit2->Text)));

For IWButton6 write the following method:

```
IWEdit1->Clear();
IWEdit2->Clear();
IWEdit3->Clear();
```

Before making the application more meaningful, compile and run it to see that it function on the web, by clearing the edit fields - button6 and try the different operations. If everything is OK, save all. Then modify the five operator buttons using the Caption, Font Size (18) and Style Properties fsBold is set to True. Make all the five operator buttons the same size by dragging the mouse around them as you did when alining and positioning them above. Mouse over the icons on the Align toolbar if you don't know how to make all buttons the same size.
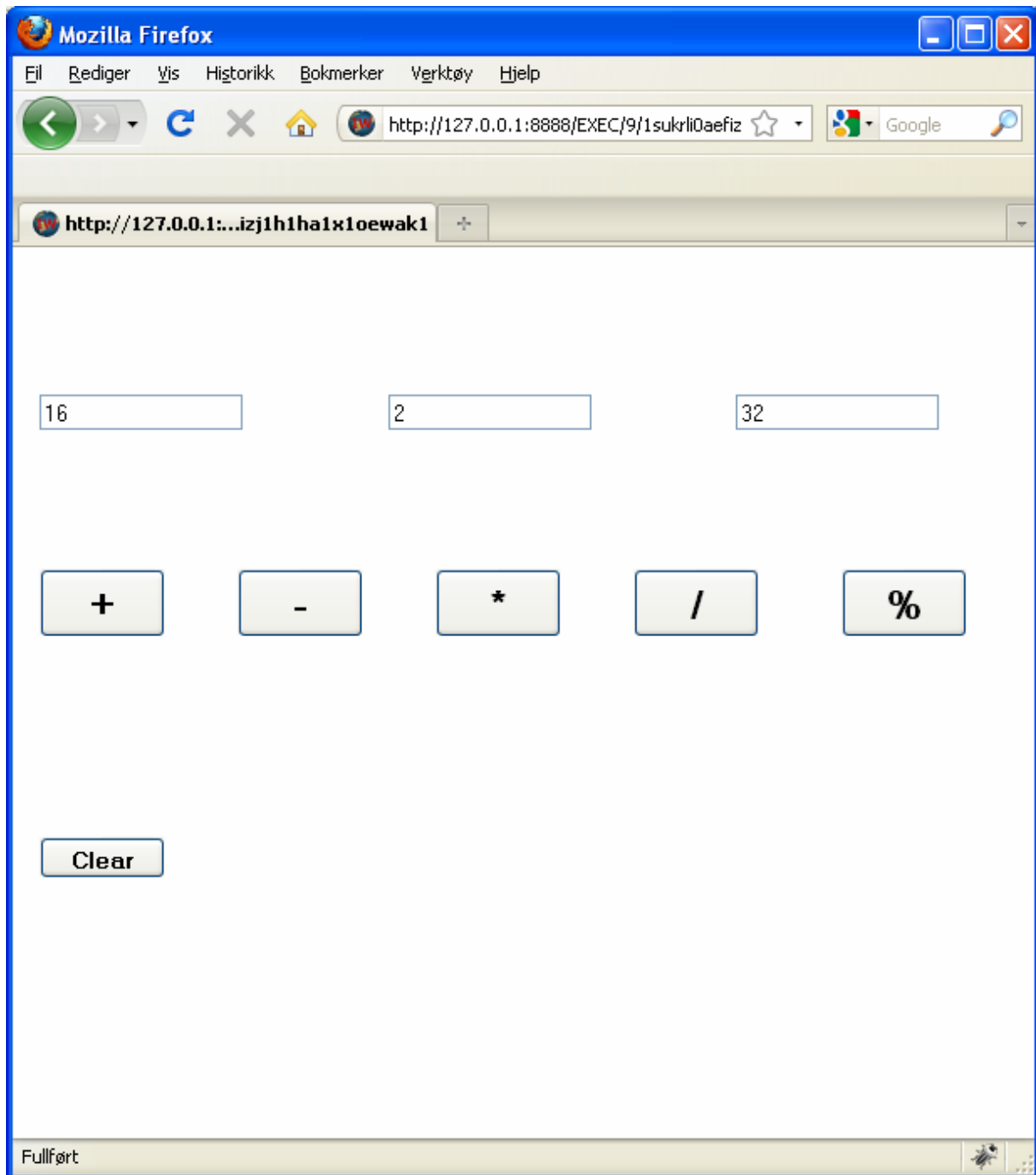
This is how the application should look on the web after you have cleared all the three edit fields, put in the numbers 16, 2 and hit multiply. Try the other buttons without changing the numbers in the edit fields. You don't need to clear the fields between the different operations.

Exercise 18.1.

1. Add a button $2^{\wedge}$ to the calculator that uses an inline assembler routine to divide and multiply by powers of 2.
2. Make a Scientific Calculator by using inbuilt functions like Sin, Exp, Ln etc.
3. Finally make the application more robust to wrong input and computation overflow:
   - Use an int type that allows computations with larger whole numbers.
   - Accept float number input like 3.14. Test the maximum size of float numbers.
   - Boundary check on input and **output** (no overflow allowed).
   - Give meaningful messages on invalid input / output. That is, check boundary conditions so now overflow is allowed and only valid input is accepted.

Exercise 18.2.

Here

Source:
ms-help://embarcadero.rs2010/rad/Creating_a_New_VCL_for_the_Web_Application.html

you can read the following:

*"The first step in the process of creating the demo program is to create a new VCL for the Web project. The project will be a stand alone application, but you can convert it to ISAPI/NSAPI **or Apache later by changing two lines of code".***

Are you able to convert the WebCalculator to an application that runs on an Apache web server?

19. The Debug toolbar. Some assembly and disassembly code.

If your debug toolbar is not active, right click on the toolbar field, activate it and put at least the View Disassembly, CPU Stack, Call Stack, CPU etc. (if you want additional training) icons on it.

On a VCL Forms Application C++ Builder, write the following form constructor method:

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
        Form1->Color=clRed;  //A form constructor event handler.
}
```

Go to the Events tab in the Object Inspector, select OnClose and double click on the right FormClose eventhandler and write in the following code:

```
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
        Close(); // This function makes trouble in C++Builder 2009 and 2010 Professional.
}
```

Compile and run the application. Try to close the form. I get this exception notification:
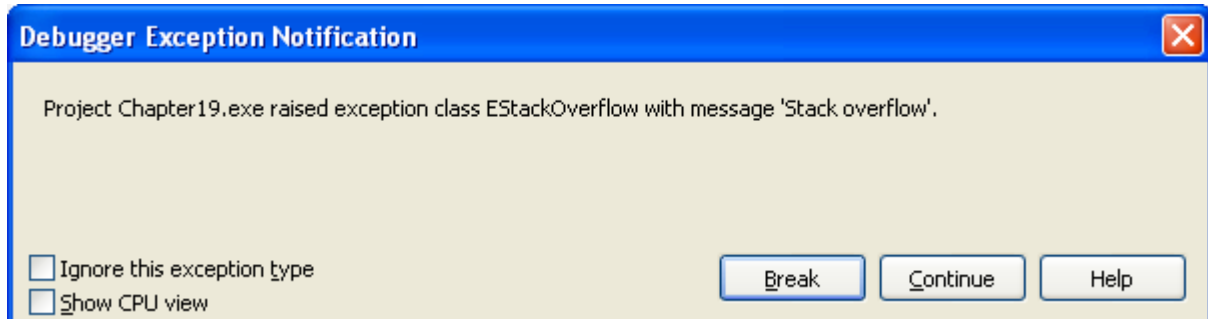
You may retry after the following experiment by checking Ignore this Exception type and / or Show CPU view, clicking Break or Help, **but now click Continue.** This happens:
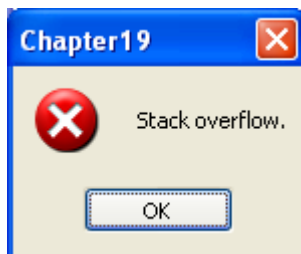
Click OK. Then try to close the application again.   (The devil is in the details. At the same time as this happened I had to change batteries in my wireless Microsoft Mouse ☺). This is the message I get.
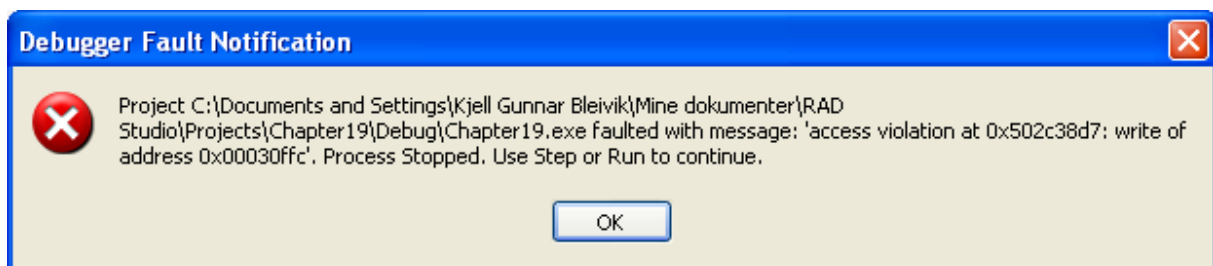

When I click OK the screen dump on the following page is shown. When I click Chapter19U.cpp, the cursor is blinking at the Close() function indicating where the error occurs. When I scroll the CPU view, bold messages like this

vcl140.@Forms@TCustomForm@CloseQuery$qqrv:

vcl140.@Forms@TCustomForm@CloseModal$qqrv:

indicate that the problem occurs when I try to close the form. **Similar messages in bold** are shown on the Disassembly View (click the Disassembly button or CTRL + Alt + D). See the

next screen dump that shows the disassembly view and it indicates that there is a problem with *push ecx*. The bolded messages that is shown when I scroll the windows indicate the error and is enough to me.
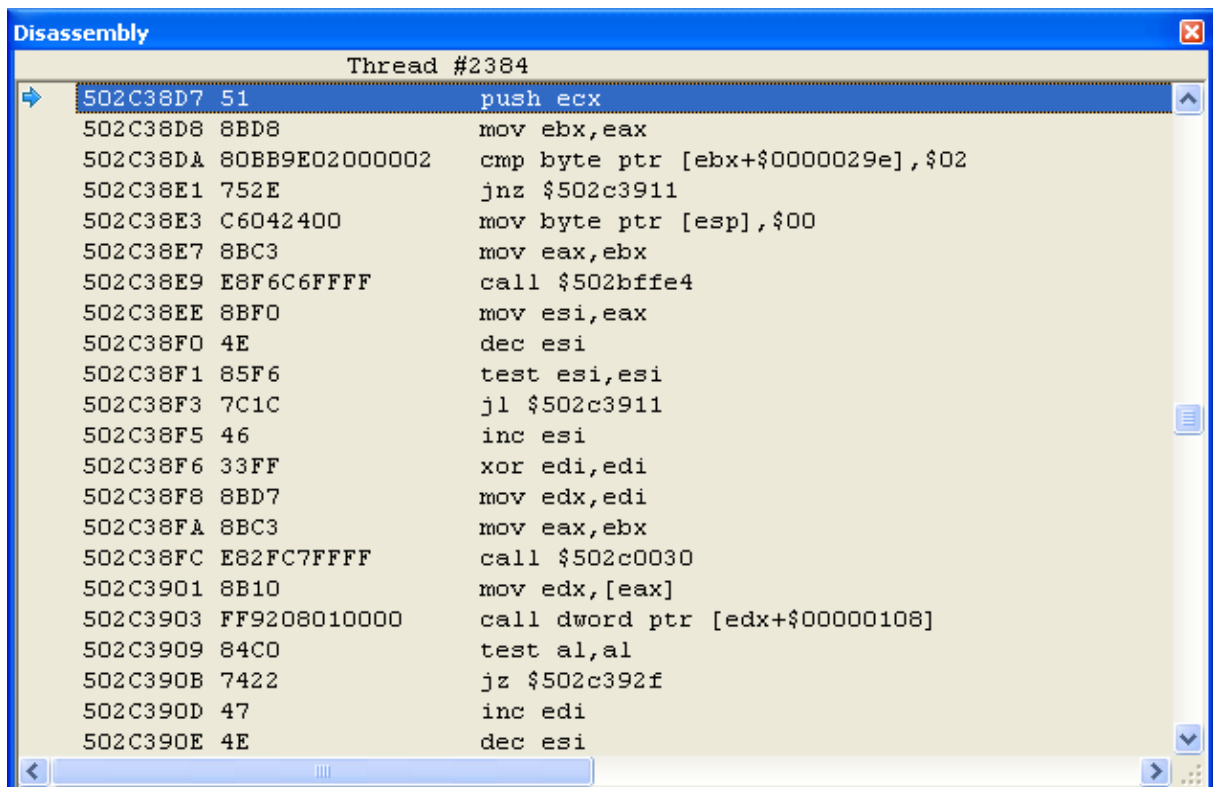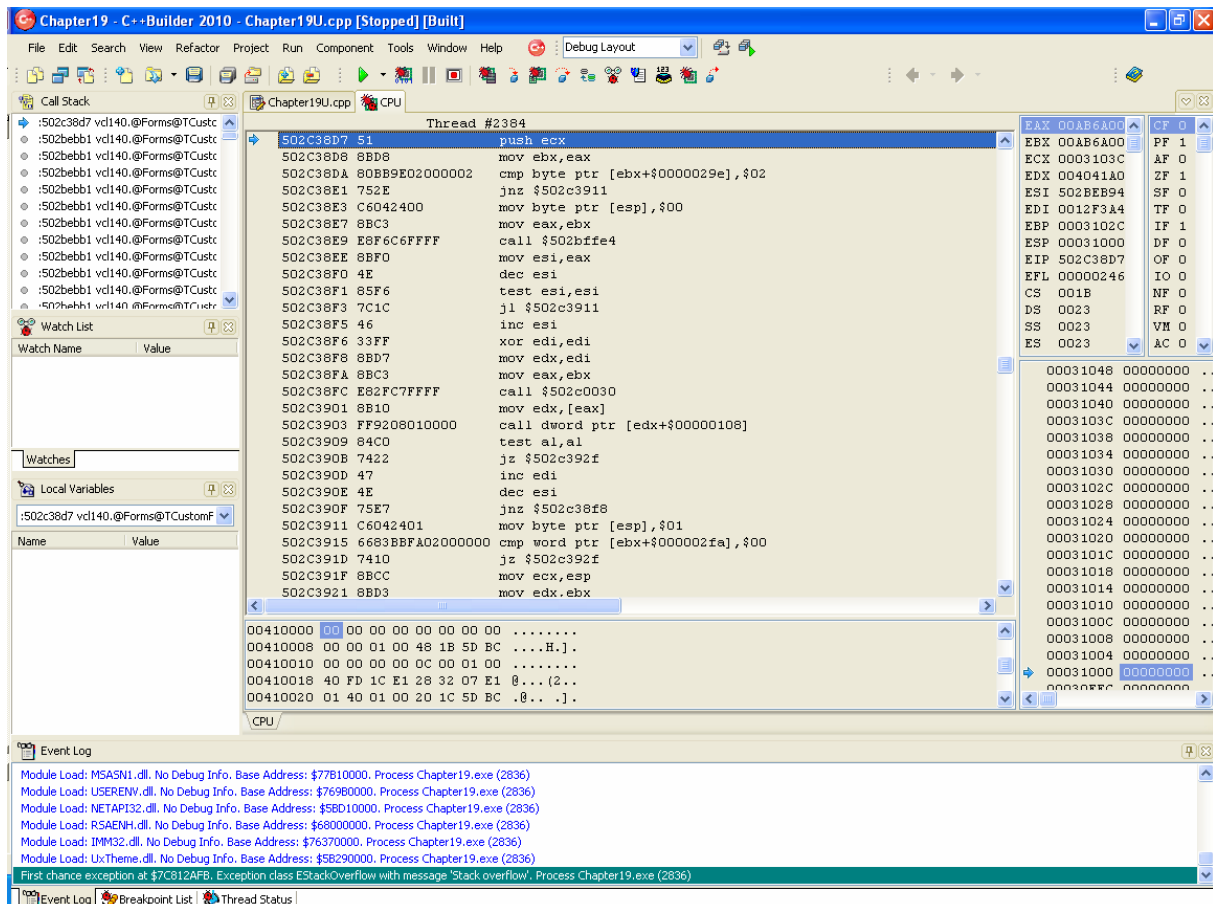
If you scroll on these two views, you can drown in assembler instructions. Play with the other buttons (acceleration keys). Unless you are an old assembler guru or a young person that want to learn (advanced debugging and) assembly Language from Square One (yes there is a book with that name http://www.kjellbleivik.com/Books/ - scroll down to the heading "Assembly programming") you can close the application. From the File menu choose "close all" or right click the program icon and choose close now. The first option makes less trouble for you. The last option can make the IDE hang in a cyclic loop so you have to close the C++Builder to continue. If you want to save the example, you can comment out:

// Close();   This function causes problems. Uncomment it and try.

Conclusion: If you end up in similar problems as explained in this simple example, there is a strong indication that there is something in your code that is wrong. Try to locate it by clicking the cpp source code file or look at bolded statements on different debug views like I did above.

Exercise 19.1.

1. In the example above, without looking at the C++ code, try to locate the error more precisely by looking at the different debug windows. That is try to find a clearer statement than we found above in the CPU and Disassemby window. I have not tried.
2. Play with the example to get a deeper understanding of your code and the functionality in the IDE.
3. Study the output by putting all available icons on the debug toolbar and play with the different buttons like we did above.

Exercise 19.2. For the deep digging student.

Study assembly in online resource and / or books to get a deeper understanding of the different debug options in the C++ Builder. You can start by looking at the link I gave you above. On that page, you find online documents and book references.

20. The sky is the limit.
I started using Borland Turbo C++ back in the beginning of 1990. I was impressed the first time I used C++ Builder 3 Professional, but the 2009 and 2010 platform has impressed me even more. Look at this

https://forums.codegear.com/message.jspa?messageID=173938#173938

"The name Delphi For PHP merely means that the product gives Delphi-like RAD capabilities to PHP. The product doesn't have anything to do with the Delphi language.

However, there are ways to integrate PHP into Delphi and C++Builder applications. The simplest way is to use the set of components called php4Delphi (http://users.telenet.be/ws36637/php4delphi.html)".

I installed Delphi PHP some minutes ago. You see a screen dump of the application on the next page. On a first look it seems impressing. Look at the toolbars. It has a jQuery toolbar, my favourite JavaScript library and a Pear toolbar, my favourite PHP library. A fast play with

some components shows that PHP code is produced behind the scenes like C++ code is produced by the C++Builder IDE.
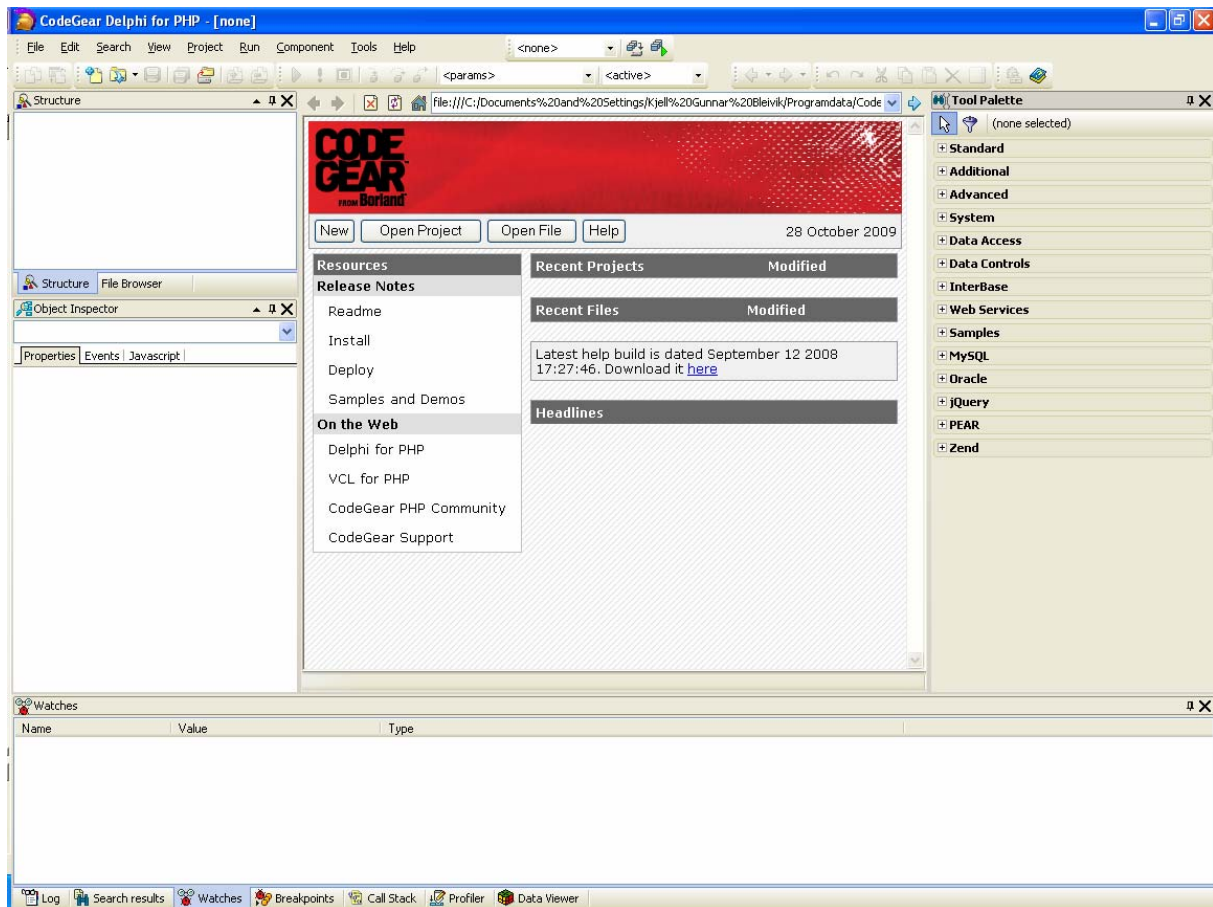
Back to C++Builder. I have experimented with the graphic tools in C++ Builder and noted that there is a lot of example code in the help index of the 2009 as well as in the 2010 version. I don't know how different the examples are. In the 2009 version, locate this page

ms-help://embarcadero.rs2009/codeexamples/examples_cpp_xml.html

Page search (CTRL + F) is an excellent way to find information on a page. Example CTRL+F

TCanvas

TGraphic

And you find a lot of examples to play with. (The class explorer should give further explanation. There is a lot of ways to view classes if you get used to the toolbar).

Exercise 20.1.
Implement the code in the two examples on this page

ms-help://embarcadero.rs2009/delphivclwin32/Graphics_GraphicExtension.html

before you look at the solution on the next page. (Note if you use Windows Vista, You can

copy and paste the code directly from the PDF document). I only solve one example for you. Read for further instructions in the comment.

```cpp
/*
This code displays a Save Picture dialog box with the
TBitmap default extension that is added automatically to
files that are given no extension. Confirm that the
bitmap file has been saved under the filename and path
specified.

Source: ms-help://embarcadero.rs2009/delphivclwin32/Graphics_GraphicExtension.html

My addition:
You must have a file factory.bmp in your project directory.
The new file must be saved in the same directory.  That is at least easiest.
*/
#include <vcl.h>
#pragma hdrstop

#include "SavePictureU.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
#include <memory>       //for STL auto_ptr class
TForm3 *Form3;
//---------------------------------------------------------------------------
__fastcall TForm3::TForm3(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm3::Button1Click(TObject *Sender)
{
  UnicodeString NewFileName, OldFileName;
  SavePictureDialog1->DefaultExt = GraphicExtension(__classid(Graphics::TBitmap));
  SavePictureDialog1->Filter = GraphicFilter(__classid(Graphics::TBitmap));
  OldFileName = "factory.bmp";
  if (SavePictureDialog1->Execute())
  {
    NewFileName = SavePictureDialog1->FileName;
    std::auto_ptr<TFileStream> OldFile(new TFileStream(OldFileName, (fmOpenRead |
fmShareDenyWrite)));
    std::auto_ptr<TFileStream> NewFile(new TFileStream(NewFileName, (fmCreate |
fmShareDenyRead)));
    NewFile->CopyFrom(OldFile.get(), OldFile->Size);
  };
}
//---------------------------------------------------------------------------
void __fastcall TForm3::FormCreate(TObject *Sender)
{
        Image1->Picture->LoadFromFile("../factory.bmp");
```

}

In C++Builder 2010, you find a similar page:

ms-help://embarcadero.rs2010/codesamples/CategoryC++.html

It may take some time until it is finished loading. Note the next 200 examples and so on.

When you are working with the graphics, it is important to get the correct coordinates. Go to the help index and look for "Coordinate Systems and Transformations". There you find that *"Microsoft Windows GDI+ uses three coordinate spaces: world, page, and device …………………………………*

Note that the page coordinate space has its origin at the upper-left corner of the client area; this will always be the case. Also note that because the unit of measure is the pixel, the device coordinates are the same as the page coordinates. If you set the unit of measure to something other than pixels (for example, inches), then the device coordinates will be different from the page coordinates."

So sometimes you have to transform from world to screen coordinates and back.

```
int x_Screen(double x, double x0, double x1, double W) {
      //Transforms x from [x0,x1] to screencoordinates [0,W]
      return (x-x0)*W/ (x1-x0);
}

int y_Screen(double y, double y0, double y1, double H) {
      //Transforms y from [y0,y1] to screencoordinates [H,0]
      return (y-y1)*H/ (y0-y1);
}

double x_World(int px, double x0, double x1, double W) {
      //Transforms px from [0,W] to Worldcoordinates [x0,x1]
      return x0 + px*(x1-x0) / W;
}

double y_World(int py, double y0, double y1, double H) {
      //Transforms py from [0,H] to Worldcoordinates [y1,y0]
      return y0 + py*(y0-y1) / H;
}
```

that may be used in functions like this:

```
void drawFunction(TCanvas* Canvas) { //Draw the function y=sin(x*x) in range [-4,4]
      double x0=-4, y0=-1, x1=4, y1=1;    //World coordinates
      Canvas->Pen->Color=clRed;
      Canvas->Pen->Width=1;

      for (int px = 0; px < Canvas->ClipRect.Width(); px++) {
            //Pixelcoordinates: px,py, Worldcoordinates: x,y
            //transform px to Worldcoordinates
```

```cpp
        double x=x_World(px,x0,x1, Canvas->ClipRect.Width());
        double y=sin(x*x);
        //transform y to pixelcoordinates
        int py = y_Screen(y,y0,y1,Canvas->ClipRect.Height());
        if (px== 0) Canvas->MoveTo(px,py);
        else Canvas->LineTo(px,py);
    }

}

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    drawFunction(Form1->Canvas);

}

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    drawFunction(Image1->Canvas);
}

void __fastcall TForm1::Button4Click(TObject *Sender)
{
    drawFunction(PaintBox1->Canvas);
}
```

see Kaisers 2007 chapter 10.13 for a further explanation.

Exercise 20.2
1. By using and modifying the above code, make a form with 6 buttons, an Image component and a PaintBox component. Make the following buttons
- DrawForm (With a similar clear button)
- DrawImage (With a clear button)
- DrawPaintBox (With a clear button)

For different mathematical functions available through #include<math.h>; draw the function on the Form, the Image and the PaintBox.
2. In the next days and weeks try to implement some of the examples that you find in the help index.
3. For those interested in Astronomy. Astro-Catalog is an astronomy module library. Here http://search.cpan.org/~aallan/Astro-Catalog-4.1.0/ is one link. Search online for

astro-Catalog C OR C++ code download

or variations thereof. The sky, excuse me, the universe is the limit. Time for some simulations? It should not be difficult if you use the correct coordinate system an can implement your own code or C and or C++ code that you find online. Even if you don't find C++ code, but Java or similar code, it should not be difficult to translate it to C++. The last time I looked at C# and Java, code it looked like simplified C++ to me personally.

21. Books, web links and other resources.

Note that there are few books on C++ Builder 2007 and later. But if you search for C++ Builder on the site: http://www.bookfinder4u.com/

you find a lot of books related to older versions of the C++ Builder. The code examples in these books may not function. Nevertheless the book may be worth the money. As an example, the book "Sams Teach Yourself Borland C++ Builder 3 in 14 Days" by Kent Reisdorph came with C++ Builder 3 Professional. I think this is a subset of the book "Sams Teach Yourself Borland C++ Builder 3 in 21 Days" by the same author. I still find the first book useful. Later books like "C++ Builder 6 Developers Guide with CDR" by Satya Sai Kolachina may be better.   Read more in this thread:
http://bcbjournal.org/forums/viewtopic.php?f=5&t=726

http://www.youtube.com/user/EmbarcaderoTechNet

http://twitter.com/EmbarcaderoTech

http://www.linkedin.com/companies/embarcadero-technologies

http://docwiki.embarcadero.com/RADStudio/en/C%2B%2B_Reference

http://blogs.embarcadero.com/chrispattinson/2009/09/22/38939

http://www.ddj.com/windows/220001058

http://www.ebob42.com/ftp/Turbo/Turbo%20C++%20Database%20Development.pdf

http://blogs.embarcadero.com/

http://blogs.embarcadero.com/chrishesik/2009/09/22/34916

http://blogs.embarcadero.com/chrispattinson/2009/09/25/38942

https://forums.codegear.com/index.jspa

https://forums.codegear.com/category.jspa?categoryID=8

https://forums.codegear.com/category.jspa?categoryID=9

https://forums.codegear.com/forum.jspa?forumID=45

https://forums.codegear.com/forum.jspa?forumID=5

http://bcbjournal.org/

http://www.lebeausoftware.org/

http://owlnext.sourceforge.net/examples.html