Course 2DC7: 2D-Computer Graphics with C++/MFC 8.0
Chapter C1: The Intro Project

An Empty Window

Start Microsoft Visual Studio 2005.
`1)` Main Menu of VS 2005: `File → New → Project...` → `Project types:` → `Visual C++` → `MFC` → `MFC Application`
`Name: intro1` → `Location: C:\temp` → `Check: Create directory for solution` → `OK`

`2)` The `MFC Application Wizard - intro1` appears.
`Application Type` → `Single document` → `No Document/View architecture support` →
`No Use Unicode libraries` → `Project Style: MFC standard` →
`Use of MFC: Use MFC in a shared DLL` → `Next>`
`Database support: None → Next>`
`Main frame styles:` → `Thick frame` → `Minimize box` → `Maximize box` → `Sytem menu` →
`No Initial status bar` → `Toolbars: None` → `Next>`
`Advanced features:` → uncheck all advanced features → `Next>`
`Generated classes: Cintro1App, CMainFrame, CChildView → Finish>`

Click `Debug` in the main menu of VS 2005. A submenu opens. Click `Start Without Debugging Ctrl F5`.
The rudimentary program now automatically compiles, links and starts. Please observe the `Error List`-window of Visual Studio below our program.
The program starts automatically as stand-alone window containing four parts:
1. main window = MainFrame with a blue title row,
2. three buttons on the right of the title bar: Minimize, Maximize, Close,
3. a main menu `File, Edit, Help`,
4. a narrow frame with 4 movable borders and 4 movable edges. Enlarge the window by dragging its borders and edges.

Minimize VisualStudio, to realize that `intro1.exe` is a stand-alone Windows program.
Start the `Explorer`. Branch to `C:\temp\intro1\bin\debug.`
Double click `intro1.exe`. You can start an arbitrary number of instances of `intro1.exe`. (You must carefully kill all of them before writing new versions.) Minimize the `Explorer`.

Make sure that all instances of `intro1.exe` have been finished.
Important: Always finish all instances of `intro1` before writing new code and starting it !
Start the Task Manager with `Ctrl+Alt+Del` and check if an `intro1.exe`-process is still to be killed.

TextOut: Hallo World

If the Solution Eyplorer - intro1 isn´t already visible, open it via the VS 2005 main menu: View → Solution Explorer, and double click the branch Source Files → ChildView.cpp.
The automatically generated source code of ChildView.cpp appears. Scroll down until You detect the message handler subroutine:

```
void CChildView::OnPaint()
{
  CPaintDC dc(this); // device context for painting

  // TODO: Add your message handler code here

  // Do not call CWnd::OnPaint() for painting messages
}
```

Delete the comments and write a new line of code until it looks like this:

```
void CChildView::OnPaint()
{ CPaintDC dc(this);
  dc.TextOut( 10, 10, "Hello world, here is intro1 !" );
}
```

Click Debug in the main menu of VS 2005. A submenu opens. Click Start Without Debugging Ctrl F5. VS 2005 compiles, links and starts the program.
Experiment: Change the first parameter of TextOut(10, ...) to TextOut(200, ...) and observe how the output is shifted to the right. Try the same with the second parameter.

Print Window Size with SetTextColor

Version 2: Finish all instances of intro1.
Write seven additional lines into void CChildView::OnPaint() until it looks like this:

```
void CChildView::OnPaint()
{ CPaintDC dc(this); // device context for painting
  dc.TextOut(10, 10, "Hello world, here is intro1 !");
  CRect r; //a rectangle r of type CRect
  CString sometext; //a string sometext of type CString
  GetClientRect( r ); //Ask the operating system to report the size of our window
  sometext.Format( "width=%d,  height=%d", r.right, r.bottom ); //prepare output
  dc.TextOut( 10, 30, sometext ); //output a line below the last one
  dc.SetTextColor(RGB(255,0,0)); //red text
  dc.TextOut(10,50, "Change the size of your window !" );
}
```

Click Debug in the main menu of VS 2005 → Start Without Debugging Ctrl F5.
Important: If there are typing errors, VS 2005 compiles nothing but displays a Message Box: There were build errors. Would you like to continue and run the last successful build ?. Quit this message box with **No**. Below the source code CChildView.cpp an Error List-window will appear. Scroll the list upwards until you see the first error. If there is not an error but a warning ignore the warning and scroll downwards until you find the first error. Double click the line reporting the first error. The cursor will automatically jump into your code to the line, where the error has been detected. Correct the typing error. Ignore all further errors (Mostly they are the consequence of the first one.) and start again and repeat this procedure until the program succeeds.
Caution: The error can occur in a command line above the cursor. Perhaps you forgot a semicolon or a bracket there.

Left, Right, Top, Bottom

Version 3: Finish all instances of `intro1`.
Write eight new lines below the already existing lines of `void CChildView::OnPaint()`:

```
  CPoint p; //a point p of type CPoint
  p.x = r.right  / 2; // mid x
  p.y = r.bottom / 2; // mid y
  dc.SetTextColor( RGB( 0, 0, 255 ) ); //blue writing
  dc.TextOut( 0         , p.y        , "left"   );
  dc.TextOut( r.right-50, p.y        , "right"  );
  dc.TextOut( p.x       , 0          , "top"    );
  dc.TextOut( p.x       , r.bottom-20, "bottom" );
```

Click `Debug` in the main menu of VS 2005 → `Start Without Debugging Ctrl F5`.

MoveTo, LineTo, Rectangle, Ellipse

Version 4: Finish all instances of `intro1`.
Write another eight new lines below the already existing lines of `void CChildView::OnPaint()`:

```
  dc.MoveTo( 0      , 0         );
  dc.LineTo( r.right, r. bottom );
  dc.MoveTo( r.right, 0         );
  dc.LineTo( 0      , r. bottom );
  int w5 = r.right  / 5; // 20% of the width
  int h5 = r.bottom / 5; // 20% of the height
  dc.Rectangle( w5, h5, 4*w5, 4*h5 );
  dc.Ellipse  ( w5, h5, 4*w5, 4*h5 );
```

Click `Debug` in the main menu of VS 2005 → `Start Without Debugging Ctrl F5`.

Draw a Star with Random Rays and Random Colors

Version 5: Finish all instances of `intro1`.
Below the three `#include` command lines on top of `CChildView.cpp` write another line starting immediately at the left margin of the text:

```
#include "math.h" //in order to find sin() and cos()
```

Write a line starting immediately at the left margin of the text directly above the `void CChildView::OnPaint()`-function

```
#define nn 120
```

Write the following lines below the already existing lines of `void CChildView::OnPaint()`:

```
  int i;
  CPen pen; //a pen of type CPen
  CPoint splash[ nn ]; //an array named splash of length nn containing x/y-coordinates
  double arcus = 2. * 3.14159 / nn; // a small segment of the unit circle
  double radius_x = 1.5 * w5; // horizontal radius of the ellipse
  double radius_y = 1.5 * h5; // vertical   radius of the ellipse
  for ( i = 0; i < nn; i++ )
  { COLORREF multicolor = RGB ( rand()%255, rand()%255, rand()%255 );
    pen.CreatePen( PS_SOLID, 20, multicolor ); //20 = thickness of the random-color pen
    dc.SelectObject( pen ); //take the pen in your hand
    double factor = (double)rand() / (double)RAND_MAX; //something between 0.0 and 1.0
    if ( factor < 0.25 ) factor = 0.25; //but not less than 1/4
    double cosinus = radius_x * factor * cos( i * arcus );
    double   sinus = radius_y * factor * sin( i * arcus );
    dc.MoveTo( p ); //mid of ellipse
    dc.LineTo( p.x + (int)cosinus, p.y + (int)sinus );//ending point
    pen.DeleteObject(); //dispose the pen
    splash[i].x = p.x + int( cosinus * 0.8 ); //store the x-coordinate
    splash[i].y = p.y + int( sinus   * 0.8  );//store the y-coordinate
  }
```

Click `Debug` in the main menu of VS 2005 → `Start Without Debugging Ctrl F5`.
Drag the borders and/or the edges of our program `intro1`.

Draw a Polygon

Version 6: Finish all instances of `intro1`.
Write the following lines below the already existing lines of `void CChildView::OnPaint()`:

```
dc.SelectStockObject( WHITE_PEN ); //an existing pen of thickness 1 named WHITE_PEN
CBrush brush; //a brush of type CBrush
brush.CreateSolidBrush( RGB( 255,0,0 ) ); //dip it into red color
dc.SelectObject( brush ); //take the brush in your hand
dc.Polygon( splash, nn ); //draw a polygon with nn vertices
brush.DeleteObject(); //dispose the brush
dc.SetTextColor( RGB( 0,0,255 ) ); //blue writing
dc.TextOut( p.x-30, p.y-8, "Splash !" );
```

Click `Debug` in the main menu of VS 2005 → `Start Without Debugging Ctrl F5`.
Drag the borders and/or the edges of our program `intro1`.

Animate it

Version 6: Finish all instances of `intro1`.
Write the following lines below the already existing lines of `void CChildView::OnPaint()`:

```
Sleep( 100 ); //slow down to 10 per second
Invalidate(); //call void CChildView::OnPaint() again
```

Click `Debug` in the main menu of VS 2005 → `Start Without Debugging Ctrl F5`.

Exercises

Read the explanations of the command lines in **Code Comments**.
Click `Help` in the main menu of VS 2005. Click the submenu `Index`.
Choose `Visual C++` in the text box `Filtered by:`. Type the following key words into `Look for:` `#define, CPoint, rand, Polygon, Ellipse, Rectangle` etc. With each key word you obtain a list.
Click the most interesting key word of the list. If there are complicated alternatives, an additional pop up window appears on the lower border of VS 2005. Read the help texts.
Finish VS 2005, start the Explorer, delete the complete directory `C:\temp\intro1`.
Start VS 2005 again and rewrite `intro1` until you can write it by heart.
Increase your writing speed by extensive use of Drag&Drop.
Invent and try new versions of the program in form of new projects `intro2, intro3` etc.
Ideas: parallel horizontal color bars, parallel vertical color bars, colorful rectangles and ellipses at random positions.