Course 2DC7: 2D-Computer Graphics with C++/MFC
Chapter C1: The Complete Code of the Intro Project

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.

Finish all instances of intro1.
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 up to the begin of ChildView.cpp. You see three #include-commands. Insert a fourth one:

```
#include "math.h"
```

Scroll down until you detect the message handler subroutine: `ChildView.cpp`
Delete the subroutine completely and replace it by the following code:

```
#define nn 120
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 !" );
  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" );
  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 );  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
  }
  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 ) );
  dc.TextOut( p.x-30, p.y-8, "Splash !" );
  Sleep( 100 ); //slow down to 10 per second
  Invalidate(); //call void CChildView::OnPaint() again
}
```

Click `Debug` and `Start Without Debugging Ctrl F5`.