

C++ TUTORIAL - UI APPLICATION USING VISUAL STUDIO 2018




(<http://www.addthis.com/bookmark.php?v=250&username=khong7>)



bogotobogo.com site search:

K Hong
google.com/+KHongSanF
Francisc...

 **Follow**

4,386 followers

Ph.D. / Golden Gate Ave, San Francisco / Seoul National Univ / Carnegie Mellon / UC Berkeley / DevOps / Deep Learning / Visualization

Sponsor Open Source development activities and free contents for everyone.



Thank you.

- K Hong (http://bogotobogo.com/about_us.php)

Application using visual studio

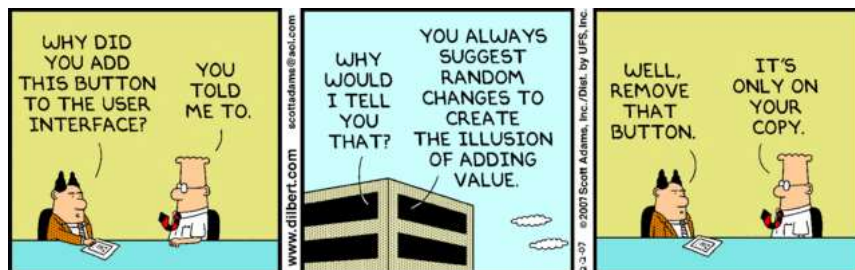
In this section, we will build **UI** application using **Windows Form** provided by Visual Studio 2013 (<http://www.microsoft.com/visualstudio/eng/2013-downloads>).



In Project Setup stage for deploy, VS 2012 will be used. Express versions will work except the project setup for deployment.

The app is a very simple random number generator with two buttons (Generator/Reset), 7 Labels for the display of the random numbers with a PictureBox.

For WPF (Windows Presentation Foundation), please visit WPF & XAML (http://www.bogotobogo.com/CSharp/csharp_wpf_xaml_netframework.php).



Source: Dilbert (<http://dilbert.com/strips/comic/2007-02-02/>)

Sponsor Open Source development activities and free contents for everyone.

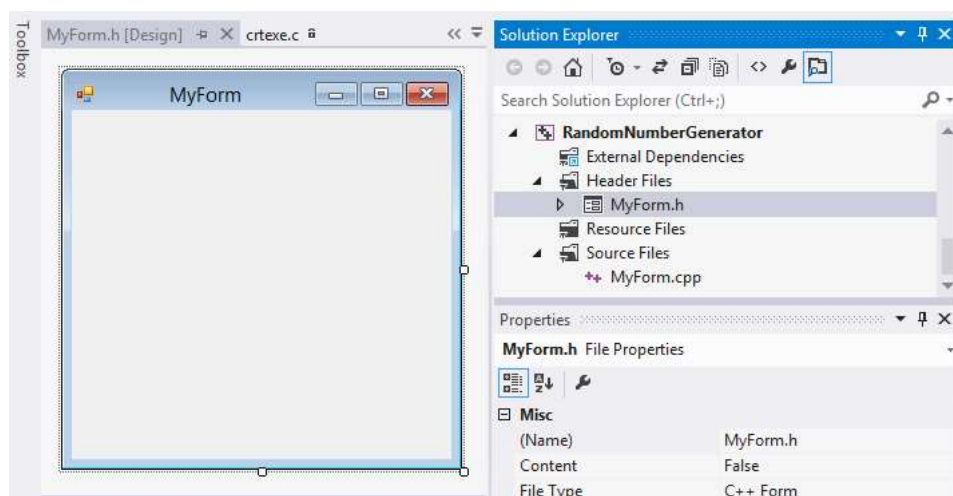
Donate with **PayPal**

Thank you.

- K Hong (http://bogotobogo.com/about_us.php)

The simplest UI program

1. Select Visual C++ **CLR** and **CLR Empty Project** and type in **RandomNumberGenerator** for the project name. The, OK.
2. **Project->Add New Item...** .
Select **UI** under **Visual C++**.
Leave the Form name as given by default **MyForm.h**.
Then, click **Add**.



3. We need to edit the **MyForm.cpp** file:

```
#include "MyForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThread]
void Main(array<String^>^ args)
{
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    RandomNumberGenerator::MyForm form;
    Application::Run(%form);
}
```

The **System** namespace provides functions to work with UI controls.

4. At the right-mouse click on **RandomNumberGenerator**, we get the **Properties** window.

Configuration Properties->Linker->System

Select **Windows (/SUBSYSTEM:WINDOWS)** for **SubSystem**.

Advanced->Entry Point, type in **Main**.

The, hit **OK**.

5. Hit **F5**, then we will have to run result, the **Form**.



C++ Tutorials

C++ Home

(/cplusplus/cpptut.php)

Algorithms & Data Structures in C++ ...

(/Algorithms/algorithms.php)

Application (UI) - using Windows Forms (Visual Studio 2013/2012)

(/cplusplus/application_visual_stu

auto_ptr

(/cplusplus/autoptr.php)

Binary Tree Example Code
(/cplusplus/binarytree.php)

Blackjack with Qt
(/cplusplus/blackjackQT.php)

Boost - shared_ptr, weak_ptr, mpl, lambda, etc.

UI Setup

1. Locate the **ToolBox**, and then expand the list of **Common Controls**.

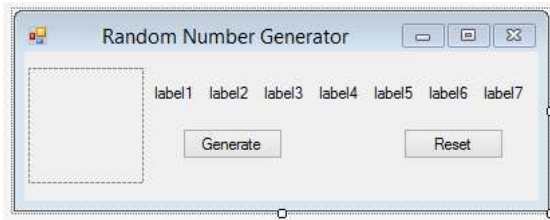
Double-click its **Label** items to add it to our **Form**.

Do this seven times.

We need to add two **Buttons** and a **PictureBox**.

Double-click those as well from the list.

2. Resize and rearrange the items. Rename the buttons and tile of the Form, then it should look like this:



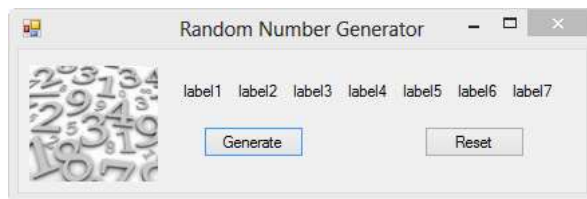
3. We can put the picture onto the PictureBox.

At a right mouse click, we get **Choosing Picture....**

Then, select the image file we want to use.

4. Let's try if it works.

Run it (Hit F5).



Event handling code for UI components

1. Let's look at the file **MyForm.h**.

(/cplusplus/boost.php)

Boost.Asio (Socket Programming - Asynchronous TCP/IP)...
(/cplusplus/Boost/boost_AsynchI

Classes and Structs
(/cplusplus/class.php)

Constructor
(/cplusplus/constructor.php)

C++11(C++0x): rvalue references, move constructor, and lambda, etc.
(/cplusplus/cplusplus11.php)

C++ API Testing
(/cplusplus/cpptesting.php)

C++ Keywords - const, volatile, etc.
(/cplusplus/cplusplus_keywords.i

Debugging Crash & Memory Leak
(/cplusplus/CppCrashDebuggingI

Design Patterns in C++ ...
(/DesignPatterns/introduction.ph

Dynamic Cast Operator
(/cplusplus/dynamic_cast.php)

Eclipse CDT / JNI (Java Native Interface) / MinGW
(/cplusplus/eclipse_CDT_JNI_MinC

Embedded Systems
Programming I - Introduction
(/cplusplus/embeddedSystemsPr

Embedded Systems
Programming II - gcc ARM
Toolchain and Simple Code on
Ubuntu and Fedora
(/cplusplus/embeddedSystemsPr

Embedded Systems
Programming III - Eclipse CDT
Plugin for gcc ARM Toolchain
(/cplusplus/embeddedSystemsPr

```
#pragma once

namespace RandomNumberGenerator {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    ///

    /// Summary for MyForm
    ///

    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void)
        {
            InitializeComponent();
            //
            //TODO: Add the constructor code here
            //
        }

        ...
    };
}
```

It begins with a **pragma once** directive.

To VS compiler, it means **only open this file once** during the compilation.

Also, as explained it before, the **System** namespace gives us functions to deal with UI controls.

The line **public ref class MyForm : public System::Windows::Forms::Form** defines a derived class named **MyForm**. The members of the class are the interface components.

2. To get a skeleton code for events, select the **Generate** button (button1), then type in **button1_Click** into for the **Click** under **Action** of the **Properties** window.

Then, VS will add additional code to **MyForm.h** for us:

```
void InitializeComponent(void)
{
    this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1_Click);
    ...

#pragma endregion
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
```

Do the same for the **Reset** button (button2).

3. Inside the bracket of **Reset** (button2), insert the following code to set the values to 0 when we click the button:

Exceptions
(/cplusplus/exceptions.php)

Friend Functions and Friend
Classes
(/cplusplus/friendclass.php)

fstream: input & output
(/cplusplus/fstream_input_output)

Function Overloading
(/cplusplus/function_overloading)

Functors (Function Objects) I -
Introduction
(/cplusplus/functor_function_obj)

Functors (Function Objects) II -
Converting function to functor
(/cplusplus/functor_function_obj)

Functors (Function Objects) -
General
(/cplusplus/functors.php)

Git and GitHub Express...
(/cplusplus/Git/Git_GitHub_Exp)

GTest (Google Unit Test) with
Visual Studio 2012
(/cplusplus/google_unit_test_gtes)

Inheritance & Virtual
Inheritance (multiple
inheritance)
(/cplusplus/multipleinheritance.p)

Libraries - Static, Shared
(Dynamic)
(/cplusplus/libraries.php)

Linked List Basics
(/cplusplus/linked_list_basics.php)

Linked List Examples
(/cplusplus/linkedlist.php)

make & CMake
(/cplusplus/make.php)

make (gnu)
(/cplusplus/gnumake.php)

```
// Reset button
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    // clear label fields
    this->label1->Text = "0";
    this->label2->Text = "0";
    this->label3->Text = "0";
    this->label4->Text = "0";
    this->label5->Text = "0";
    this->label6->Text = "0";
    this->label7->Text = "0";

    // set button state
    this->button1->Enabled = true;
    this->button2->Enabled = false;
}
```

Also, the fields should be set to 0 when we load the form. So, click the **Label1**, then set the **Text** to **0** under **Properties** window. Repeat the same to the reset of the labels. Note that we disabled the **Reset** button, and enabled the **Generate** button at the click.



Generate Random numbers

1. When the **Generate** is clicked, random numbers should be generated and displayed. We will put the code into the event handling function,

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e).
```

Memory Allocation

(/cplusplus/memoryallocation.ph

Multi-Threaded Programming - Terminology - Semaphore, Mutex, Priority Inversion etc. (/cplusplus/multithreaded.php)

Multi-Threaded Programming II - Native Thread for Win32 (A) (/cplusplus/multithreading_win32

Multi-Threaded Programming II - Native Thread for Win32 (B) (/cplusplus/multithreading_win32

Multi-Threaded Programming II - Native Thread for Win32 (C) (/cplusplus/multithreading_win32

Multi-Threaded Programming II - C++ Thread for Win32 (/cplusplus/multithreading_win32

Multi-Threaded Programming III - C/C++ Class Thread for Pthreads (/cplusplus/multithreading_pthre

MultiThreading/Parallel Programming - IPC (/cplusplus/multithreading_ipc.pl

Multi-Threaded Programming with C++11 Part A (start, join(), detach(), and ownership) (/cplusplus/multithreaded4_cplu

Multi-Threaded Programming with C++11 Part B (Sharing Data - mutex, and race conditions, and deadlock) (/cplusplus/multithreaded4_cplu

Multithread Debugging (/cplusplus/multithreadedDebug

Object Returning (/cplusplus/object_returning.php

Object Slicing and Virtual Table (/cplusplus/slicing.php)

```
// Generate button
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {

    int num[7] = { 0 };

    // seed
    srand((int) time(0));

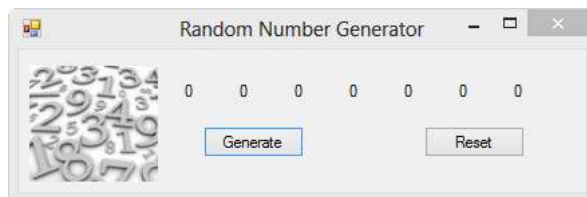
    // Randomize the array values.
    for (int i = 0; i < 7; i++)
        num[i] = (rand() % 99) + 1;

    // set the label text with random number
    this->label1->Text = Convert::ToString(num[0]);
    this->label2->Text = Convert::ToString(num[1]);
    this->label3->Text = Convert::ToString(num[2]);
    this->label4->Text = Convert::ToString(num[3]);
    this->label5->Text = Convert::ToString(num[4]);
    this->label6->Text = Convert::ToString(num[5]);
    this->label7->Text = Convert::ToString(num[6]);

    // change the button states.
    this->button1->Enabled = false;
    this->button2->Enabled = true;
}
```

For more info on the random number, please visit Random Numbers in C++
(<http://www.bogotobogo.com/cplusplus/RandomNumbers.php>).

2. Press **F5** to run it again.



Deploy

1. Launch the **Configuration Manager...**, and select **Release** from **Active solution configuration**.

OpenCV with C++
(</cplusplus/opencv.php>)

Operator Overloading I
(</cplusplus/operatoroverloading>).

Operator Overloading II - self assignment
(/cplusplus/operator_oveloding).

Pass by Value vs. Pass by Reference
(</cplusplus/valuevsreference.php>)

Pointers
(</cplusplus/pointers.php>)

Pointers II - void pointers & arrays
(/cplusplus/pointers2_voidpointe)

Pointers III - pointer to function & multi-dimensional arrays
(/cplusplus/pointers3_function_m)

Preprocessor - Macro
(/cplusplus/preprocessor_macro).

Private Inheritance
(/cplusplus/private_inheritance.p)

Python & C++ with SIP
(/python/python_cpp_sip.php)

(Pseudo)-random numbers in C++
(</cplusplus/RandomNumbers.ph>)

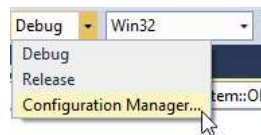
References for Built-in Types
(</cplusplus/references.php>)

Socket - Server & Client
(/cplusplus/sockets_server_client)

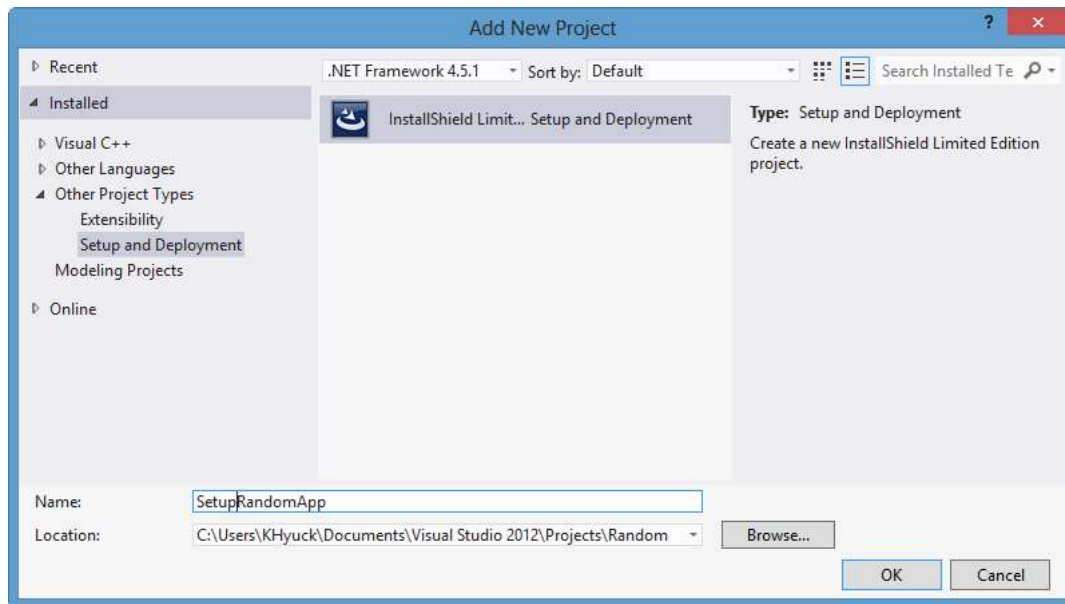
Socket - Server & Client with Qt (Asynchronous / Multithreading / ThreadPool etc.)
(/cplusplus/sockets_server_client)

Stack Unwinding
(</cplusplus/stackunwinding.php>)

Standard Template Library (STL) I - Vector & List



2. We've done the following steps for the **Debug** version. Now, let's do it for **Release** version.
At the right-mouse click on **RandomNumberGenerator**, we get the **Properties** window.
Configuration Properties->Linker->System
Select **Windows (/SUBSYSTEM:WINDOWS)** for **SubSystem**.
Advanced->Entry Point, type in **Main**.
The, hit **OK**.
3. To deploy the application, a **Setup Project** should be added to the solution to create the required files for installation.
4. **File->New**. Launch a **New Project** dialog.
5. From the **New Project** dialog, choose **Other Project Types->Setup and Deployment**.
We need to enter a name for the **Setup Project**.



Click **OK** to add the project files to the **Solution**. Then we see the **SetupProject** in the Solution Explorer.

(/cplusplus/stl_vector_list.php)

Standard Template Library
(STL) II - Maps
(/cplusplus/stl2_map.php)

Standard Template Library
(STL) II - unordered_map
(/cplusplus/stl2_unorderd_map_c

Standard Template Library
(STL) II - Sets
(/cplusplus/stl2B_set.php)

Standard Template Library
(STL) III - Iterators
(/cplusplus/stl3_iterators.php)

Standard Template Library
(STL) IV - Algorithms
(/cplusplus/stl4_algorithms.php)

Standard Template Library
(STL) V - Function Objects
(/cplusplus/stl5_function_objects

Static Variables and Static Class
Members
(/cplusplus/statics.php)

String (/cplusplus/string.php)

String II - stringstream etc.
(/cplusplus/string2.php)

Taste of Assembly
(/cplusplus/assembly.php)

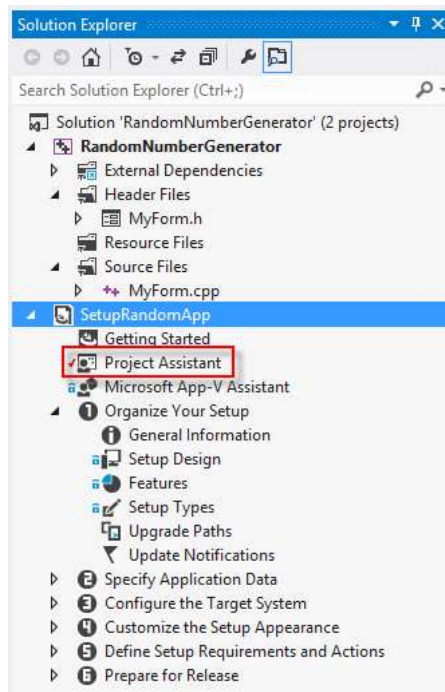
Templates
(/cplusplus/templates.php)

Template Specialization
(/cplusplus/template_specializati

Template Specialization - Traits
(/cplusplus/template_specializati

Template Implementation &
Compiler (.h or .cpp?)
(/cplusplus/template_declaration

The this Pointer
(/cplusplus/this_pointer.php)



Type Cast Operators
(/cplusplus/typecast.php)

Upcasting and Downcasting
(/cplusplus/upcasting_downcasti

Virtual Destructor &
boost::shared_ptr
(/cplusplus/virtual_destructors_sl

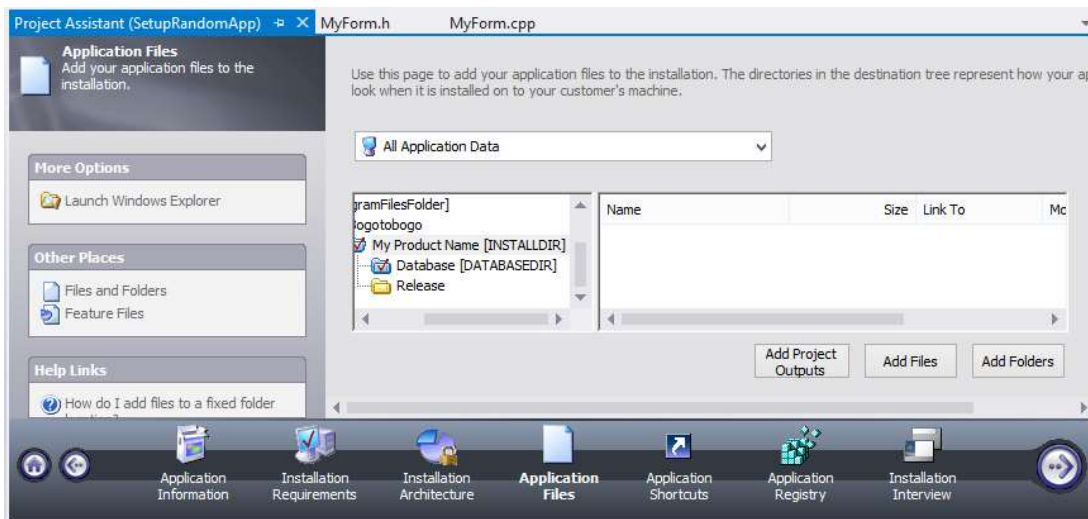
Virtual Functions
(/cplusplus/virtualfunctions.php)

Programming Questions and Solutions ↓

Strings and Arrays
(/cplusplus/quiz_strings_arrays.p

Linked List
(/cplusplus/quiz_linkedlist.php)

6. From the **Project Assistant** window, we setup the properties of installation.



Recursion
(/cplusplus/quiz_recursion.php)

Bit Manipulation
(/cplusplus/quiz_bit_manipulation

Small Programs (string,
memory functions etc.)
(/cplusplus/smallprograms.php)

Math & Probability
(/cplusplus/quiz_math_probabil

Multithreading
(/cplusplus/quiz_multithreading.p

140 Questions by Google
(/cplusplus/google_interview_que

For example, the picture shows adding a **Release** folder to the install.

7. After setup the install, we build the **Setup Project**. In this case, we do right click on the **SetupRandomApp** in the Solution Explorer

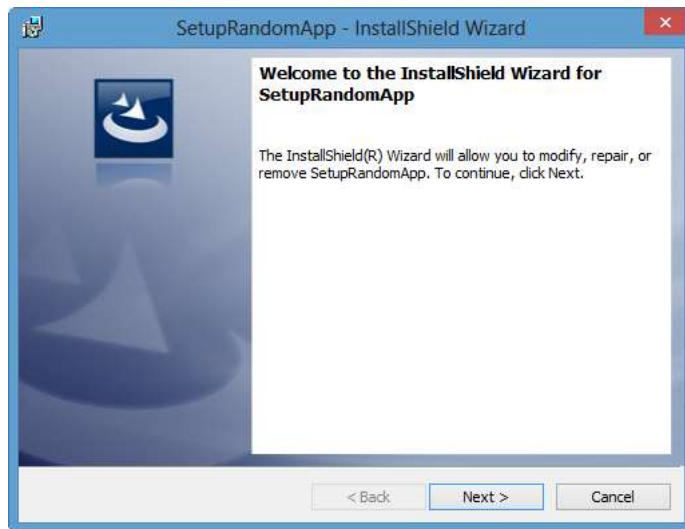
8. Then, locate the **setup.exe** file and run. In this case, it's in

C:\Users\KHyuck\Documents\Visual Studio 2012\Projects\Random\SetupRandomApp\SetupRandomApp\Express\SingleImage\DiskImages\DISK1

Qt 5 EXPRESS...
(/Qt/Qt5_Creating_QtQuick2_QM

Win32 DLL ...
(/Win32API/Win32API_DLL.php)

Articles On C++
(/cplusplus/cppNews.php)



What's new in C++11...
(/cplusplus/C11/C11_initializer_lis

C++11 Threads EXPRESS...
(/cplusplus/C11/1_C11_creating_t

OpenCV...
(/OpenCV/opencv_3_tutorial_img

9. Go to the install directory, run the **RandomNumberGenerator.exe**. In this example, It's installed in **C:\Program Files (x86)\Bogotobogo\My Product Name\Release** directory.

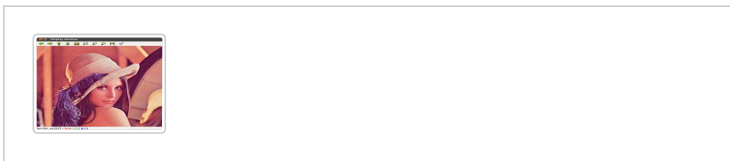
Source Files

Source files used in the example is
(<http://www.bogotobogo.com/cplusplus/files/application/Random.zip>)Random.zip.

Bogotobogo Image / Video Processing

Computer Vision & Machine Learning

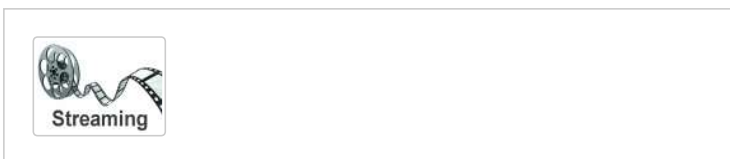
with OpenCV, MATLAB, FFmpeg, and scikit-learn.



I hope this site is informative and helpful. (/about_us.php)

Bogotobogo's Video Streaming Technology

with FFmpeg, HLS, MPEG-DASH, H.265 (HEVC)



I hope this site is informative and helpful. (/about_us.php)

Bogotobogo's contents

To see more items, click left or right arrow.



I hope this site is informative and helpful. (/about_us.php)

CONTACT

BogoToBogo
contactus@bogotobogo.com (mailto: #)

FOLLOW BOGOTOBOGO

f (<https://www.facebook.com/KHongSanFrancisco>) **t**
(<https://twitter.com/KHongTwit>) **g+**
(<https://plus.google.com/u/0/+KHongSanFrancisco/posts>)

ABOUT US (/ABOUT_US.PHP)

contactus@bogotobogo.com (mailto: #)

Golden Gate Ave, San Francisco, CA 94115

Golden Gate Ave, San Francisco, CA 94115