Lesson 1:
The OpenCV Libraries

Arturo de la Escalera & Jorge Beltrán
Course 2019-2020

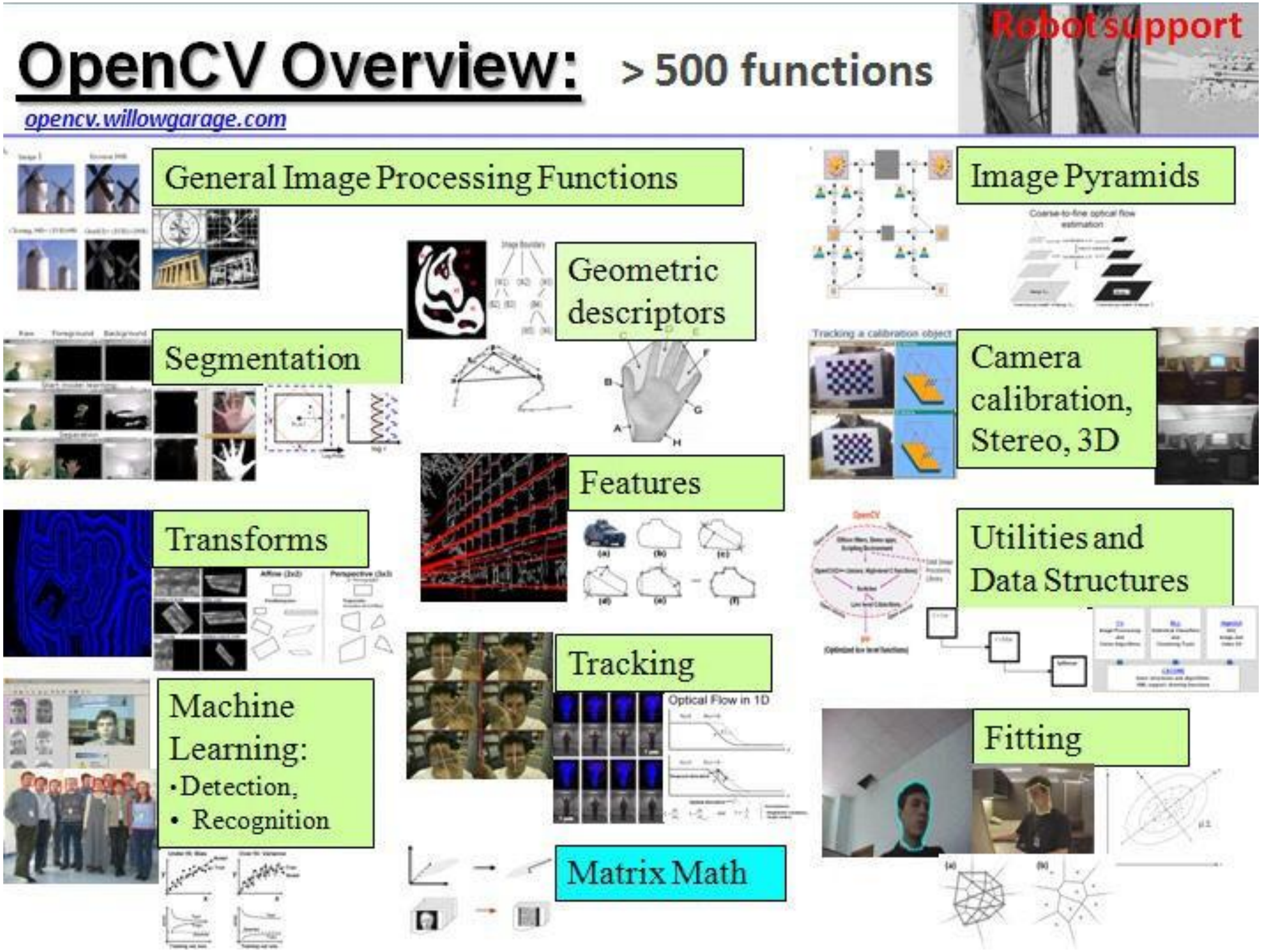
Lesson 1: The OpenCV Libraries

¿What is OpenCV?

- Open Source Computer Vision
- Computer vision libraries developed by Intel
- 1999 alfa version, 2018 4.0 versión
- BSD license. It can be used for commercial and research purposes
- Multiplatform: Linux, MacOS X & Windows
- +500 functions (C, C++, Python)

- Web: **opencv.org**

Lesson 1: The OpenCV Libraries



OpenCV Overview: > 500 functions

opencv.willowgarage.com

Robot support

General Image Processing Functions

Segmentation

Transforms

Machine Learning:
- Detection,
- Recognition

Geometric descriptors

Features

Tracking

Matrix Math

Image Pyramids

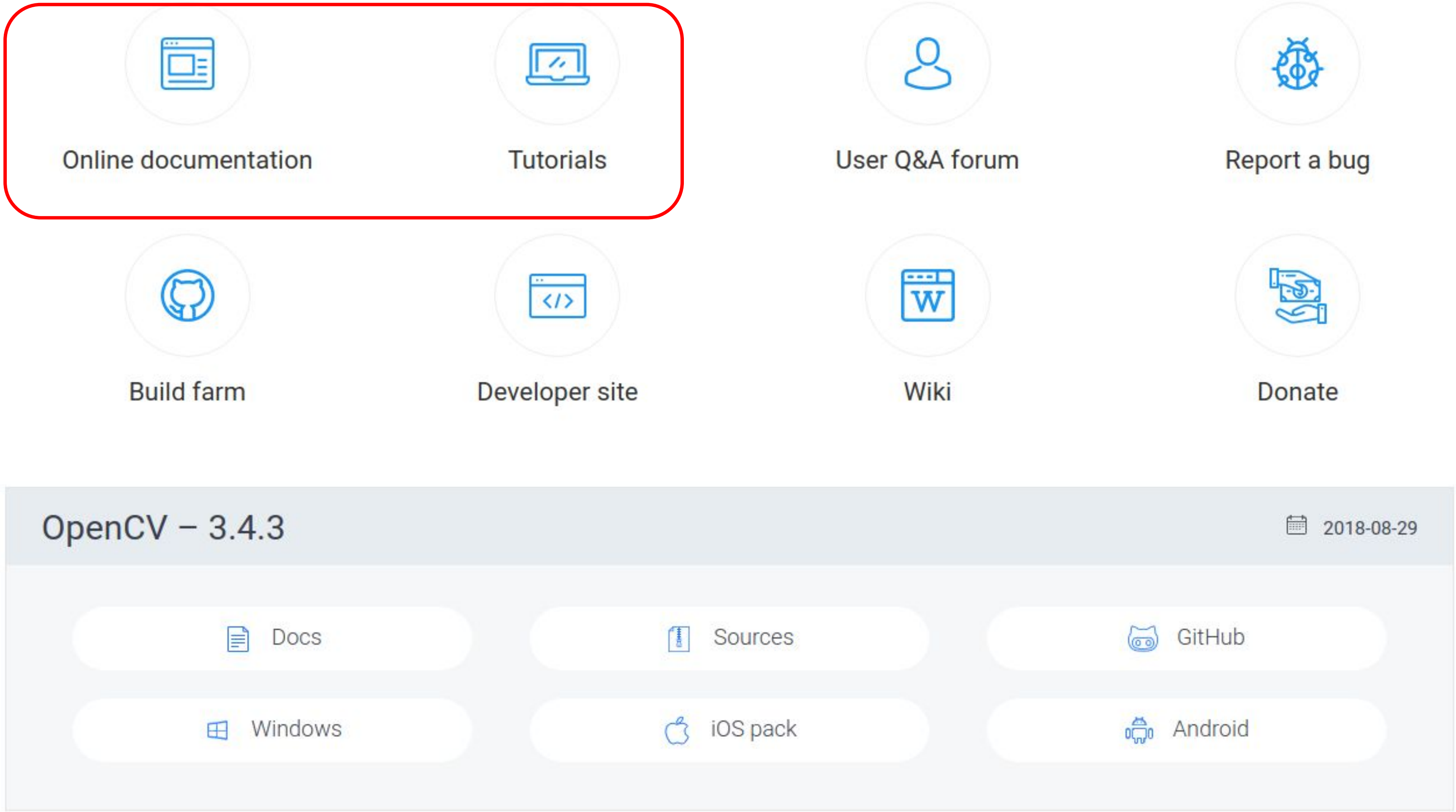
Camera calibration, Stereo, 3D

Utilities and Data Structures

Fitting

Coarse-to-fine optical flow estimation
Tracking a calibration object
Optical Flow in 1D
OpenCV
Affine (2x2)
Perspective (3x3)

Lesson 1: The OpenCV Libraries



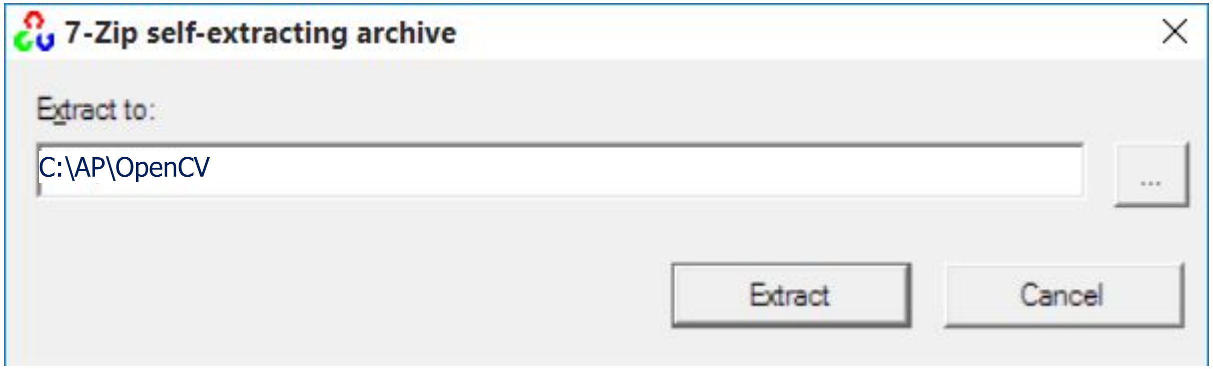
http://opencv.org

Lesson 1: The OpenCV Libraries

Installation steps

- Download from: https://opencv.org/releases/page/2/ Version 3.4.3 (August, 2018)

- Install in: C:\AP\OpenCV (or C:\opencv in your PC)



C:\AP\OpenCV

Lesson 1: The OpenCV Libraries

In Microsoft Visual Studio:

> New project

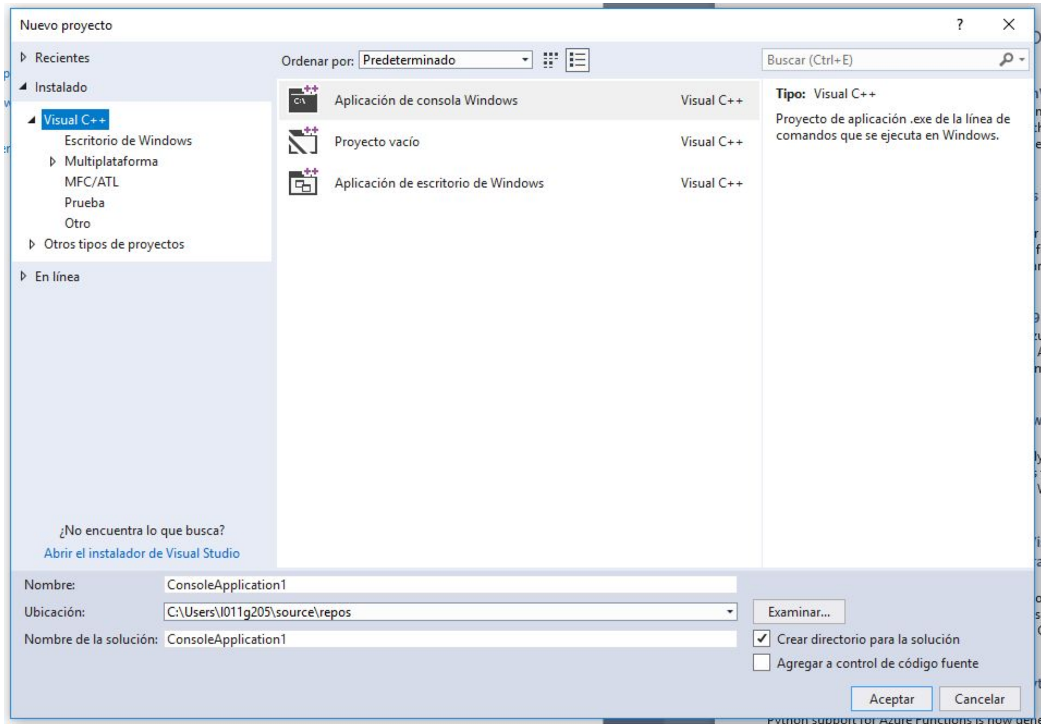
> Add subdirectories and libraries path

> Select the Project libraries

How?

Lesson 1: The OpenCV Libraries

- Open Microsoft Visual Studio
  File > New > Project (Archivo > Nuevo > Proyecto)
- Choose project type:
  Visual C++ -> Win32 Console Application (Aplicación de consola Windows)
- Name -> Project name



Nuevo proyecto

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Aplicación de consola Windows	Visual C++
Proyecto vacío	Visual C++
Aplicación de escritorio de Windows	Visual C++

Tipo: Visual C++

Proyecto de aplicación .exe de la línea de comandos que se ejecuta en Windows.

¿No encuentra lo que busca?

Abrir el instalador de Visual Studio

Nombre: ConsoleApplication1

Ubicación: C:\Users\I011g205\source\repos

Nombre de la solución: ConsoleApplication1

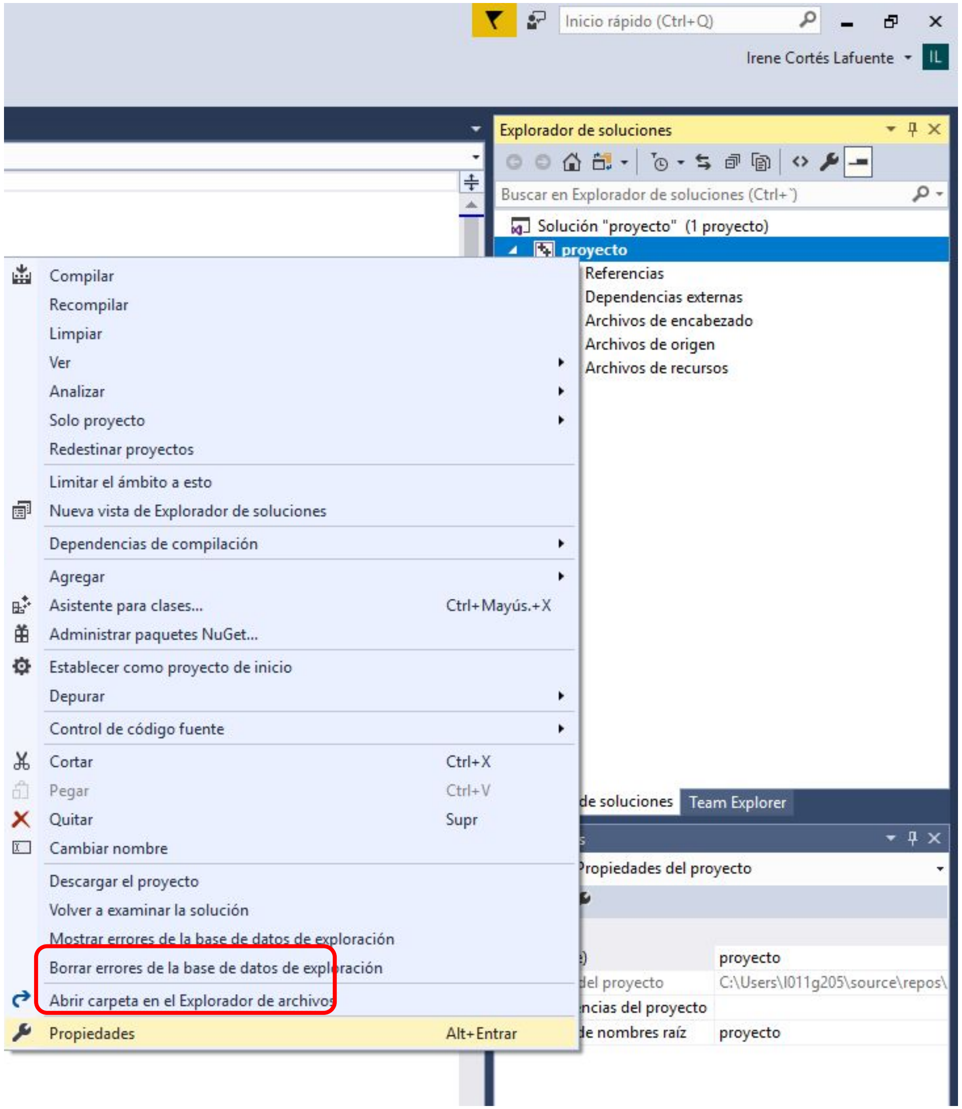
☑ Crear directorio para la solución

□ Agregar a control de código fuente

Lesson 1: The OpenCV Libraries

Include subdirectories and libraries path.

Right click on Project name to open the pop-up menu. Then, select *Properties.*



Lesson 1: The OpenCV Libraries

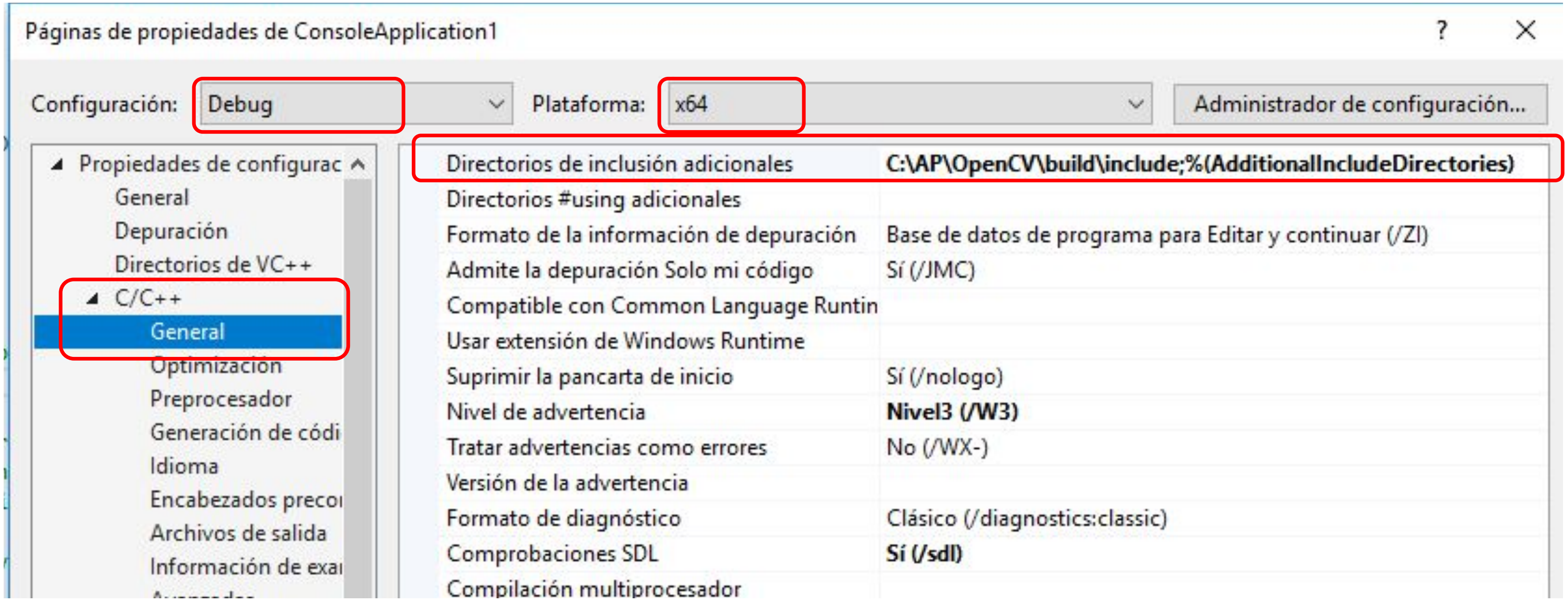
In the upper bar

Configuration (Configuración) > All configurations (Todas las configuraciones)

Platform (Plataforma) > x64

In C/C++ > General

Include directories (Directorios de inclusión adicionales): C:\AP\OpenCV\build\include



Configuración: Debug

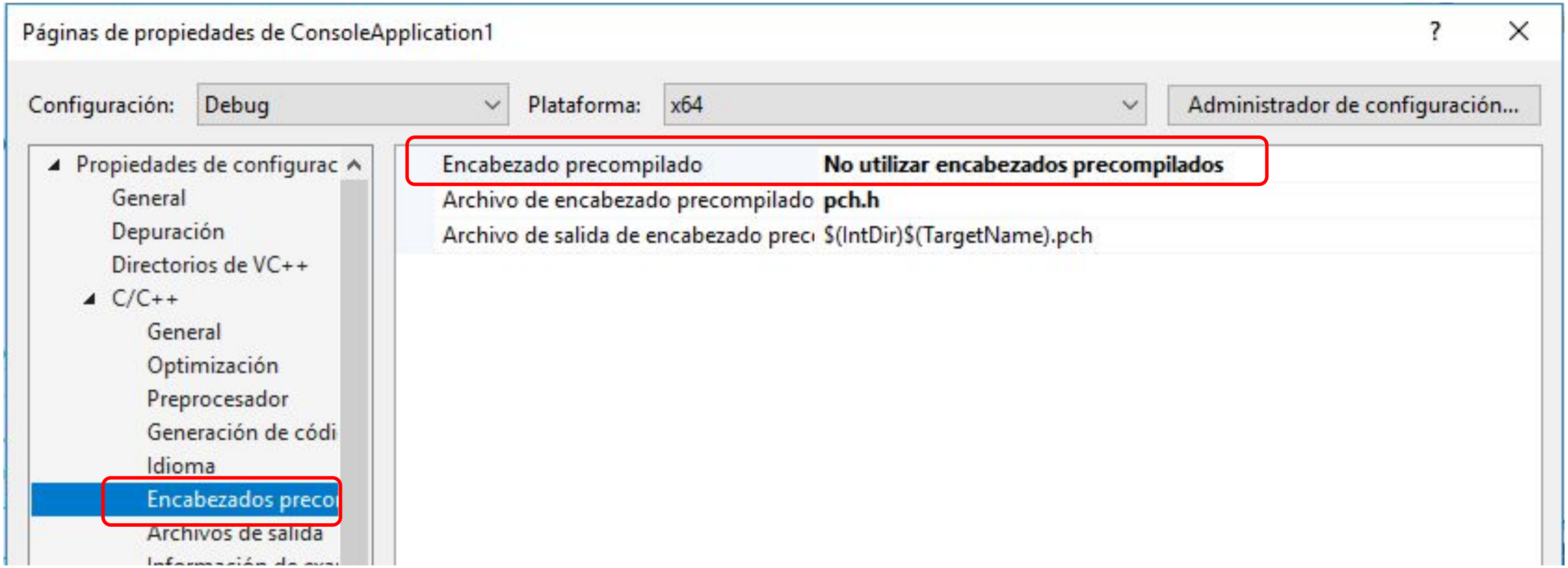
Plataforma: x64

Directorios de inclusión adicionales	**C:\AP\OpenCV\build\include;%(AdditionalIncludeDirectories)**
Directorios #using adicionales	
Formato de la información de depuración	Base de datos de programa para Editar y continuar (/ZI)
Admite la depuración Solo mi código	Sí (/JMC)
Compatible con Common Language Runtin	
Usar extensión de Windows Runtime	
Suprimir la pancarta de inicio	Sí (/nologo)
Nivel de advertencia	**Nivel3 (/W3)**
Tratar advertencias como errores	No (/WX-)
Versión de la advertencia	
Formato de diagnóstico	Clásico (/diagnostics:classic)
Comprobaciones SDL	**Sí (/sdl)**
Compilación multiprocesador	

Lesson 1: The OpenCV Libraries

In C/C++ > Precompiled headers (Encabezados precompilados)

Precompiled header (Encabezado precompilado) > No (No utilizar encabezados precompilados)



Configuración: Debug

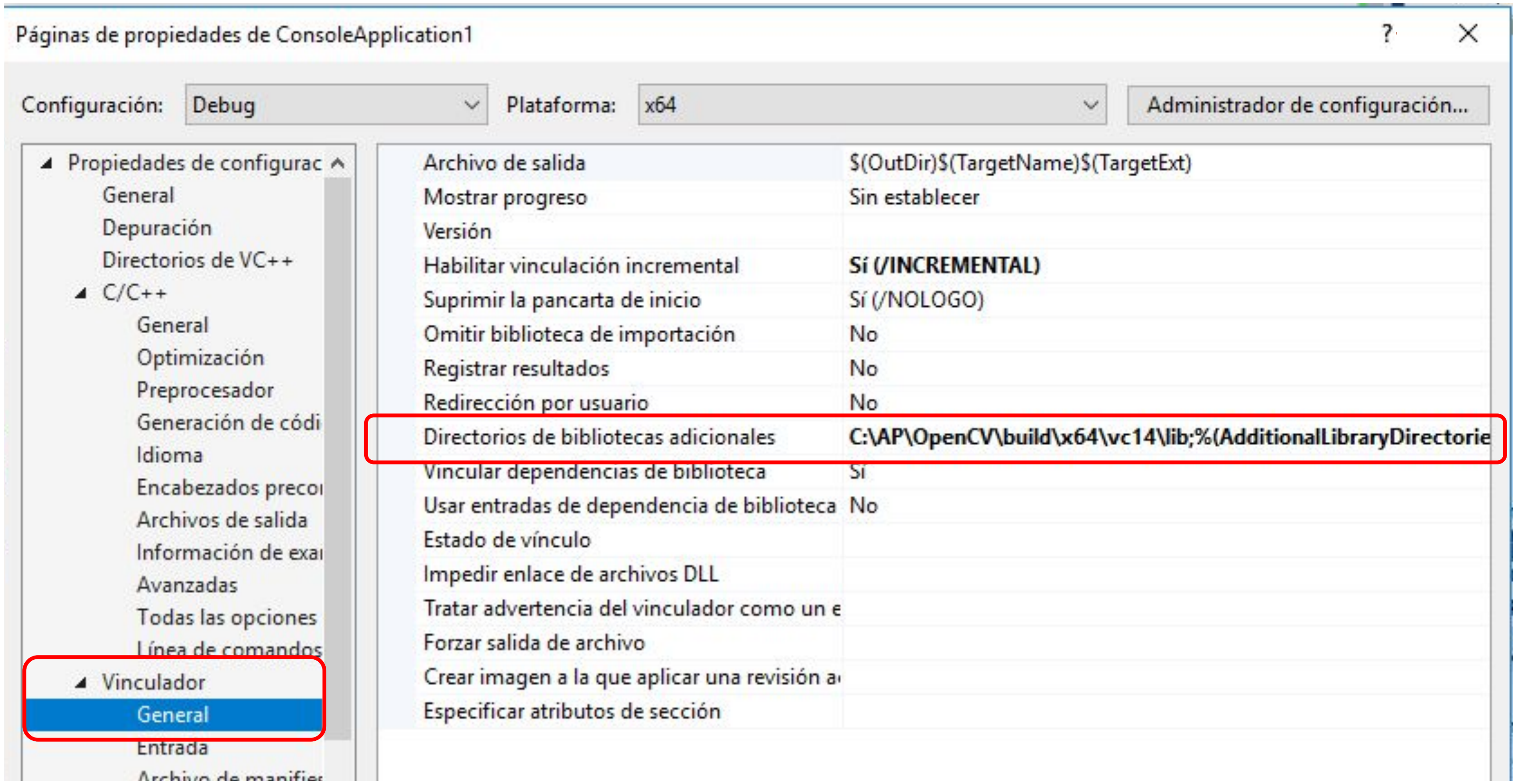
Plataforma: x64

Encabezado precompilado	**No utilizar encabezados precompilados**
Archivo de encabezado precompilado	**pch.h**
Archivo de salida de encabezado preci	$(IntDir)$(TargetName).pch

Lesson 1: The OpenCV Libraries

Linker (Vinculador) > General

Library Directories (Directorios de bibliotecas adicionales): C:\AP\OpenCV\build\x64\vc14\lib



Páginas de propiedades de ConsoleApplication1

Configuración: Debug

Plataforma: x64

Archivo de salida	$(OutDir)$(TargetName)$(TargetExt)
Mostrar progreso	Sin establecer
Versión	
Habilitar vinculación incremental	**Sí (/INCREMENTAL)**
Suprimir la pancarta de inicio	Sí (/NOLOGO)
Omitir biblioteca de importación	No
Registrar resultados	No
Redirección por usuario	No
Directorios de bibliotecas adicionales	**C:\AP\OpenCV\build\x64\vc14\lib;%(AdditionalLibraryDirectorie**
Vincular dependencias de biblioteca	Sí
Usar entradas de dependencia de biblioteca	No
Estado de vínculo	
Impedir enlace de archivos DLL	
Tratar advertencia del vinculador como un e	
Forzar salida de archivo	
Crear imagen a la que aplicar una revisión a	
Especificar atributos de sección	

Lesson 1: The OpenCV Libraries

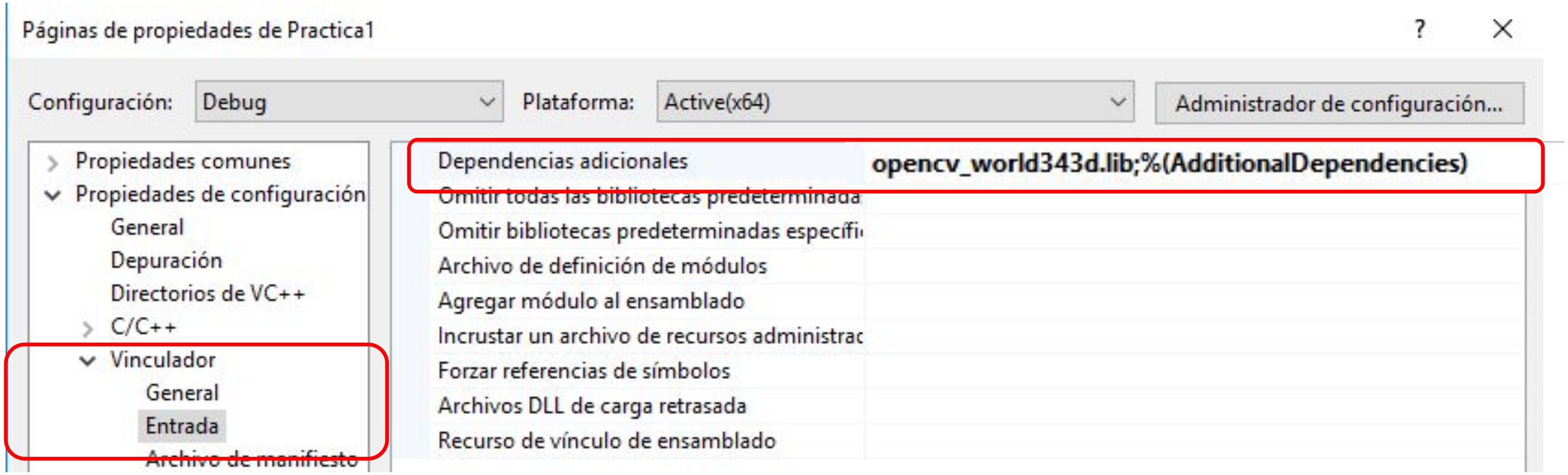
Link OpenCV libraries to the project:

Configuration: Debug | Release

Linker (Vinculador) > Input (Entrada)

Additional dependencies (Dependencias adicionales):

- opencv_world343d.lib | opencv_world343.lib



Páginas de propiedades de Practica1

Configuración: Debug

Plataforma: Active(x64)

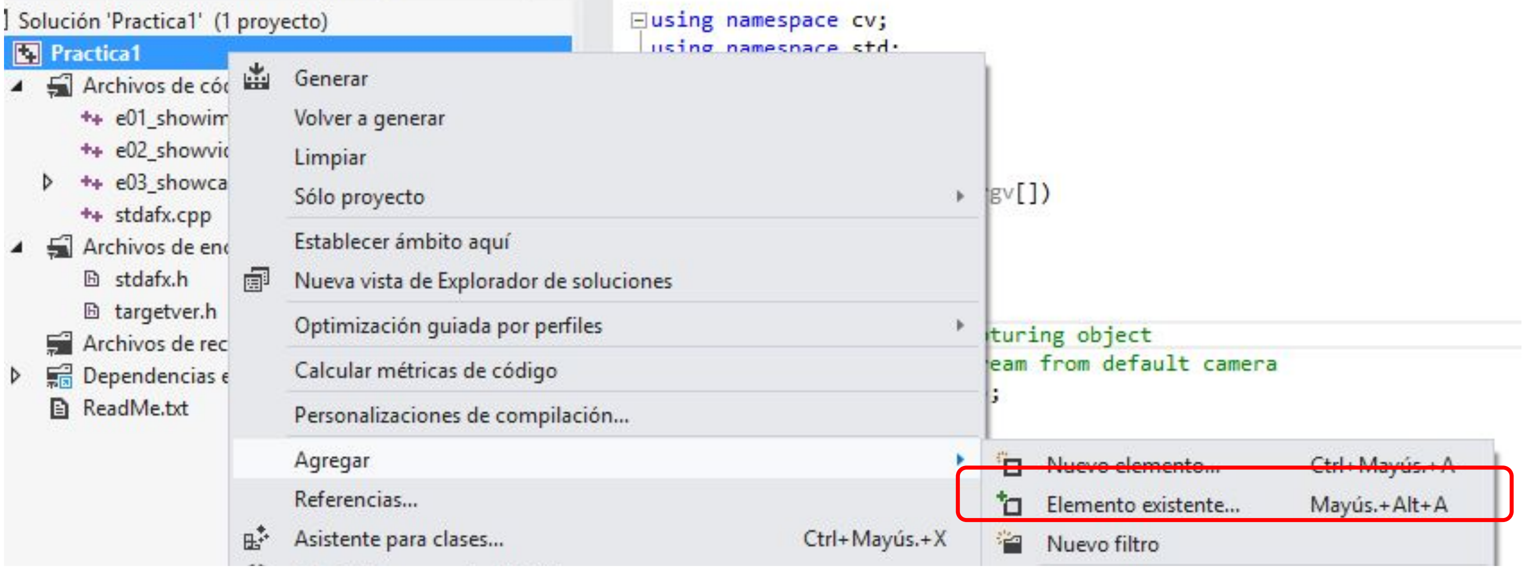
Dependencias adicionales: opencv_world343d.lib;%(AdditionalDependencies)

?
Administrador de configuración...
Propiedades comunes
Propiedades de configuración
General
Depuración
Directorios de VC++
C/C++
Vinculador
General
Entrada
Archivo de manifiesto
Omitir todas las bibliotecas predeterminada
Omitir bibliotecas predeterminadas específi
Archivo de definición de módulos
Agregar módulo al ensamblado
Incrustar un archivo de recursos administrac
Forzar referencias de símbolos
Archivos DLL de carga retrasada
Recurso de vínculo de ensamblado

Lesson 1: The OpenCV Libraries

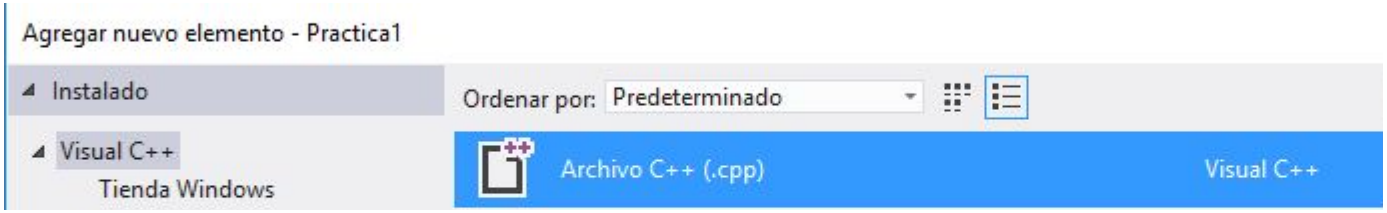
To add a new source code files

- Right click on the project name > Add (Agregar) > New element (Nuevo elemento)


Solución 'Practica1' (1 proyecto)
Practica1
Archivos de cód
e01_showim
e02_showvic
e03_showca
stdafx.cpp
Archivos de enc
stdafx.h
targetver.h
Archivos de rec
Dependencias e
ReadMe.txt
Generar
Volver a generar
Limpiar
Sólo proyecto
Establecer ámbito aquí
Nueva vista de Explorador de soluciones
Optimización guiada por perfiles
Calcular métricas de código
Personalizaciones de compilación...
Agregar
Referencias...
Asistente para clases... Ctrl+Mayús.+X
Nuevo elemento... Ctrl+Mayús.+A
Elemento existente... Mayús.+Alt+A
Nuevo filtro

```
using namespace cv;
using namespace std;
gv[])
turing object
eam from default camera
;
```

- Visual C++ > C++ File .cpp (Archivo C++ .cpp)


Agregar nuevo elemento - Practica1
Instalado
Visual C++
Tienda Windows
Ordenar por: Predeterminado
Archivo C++ (.cpp)
Visual C++

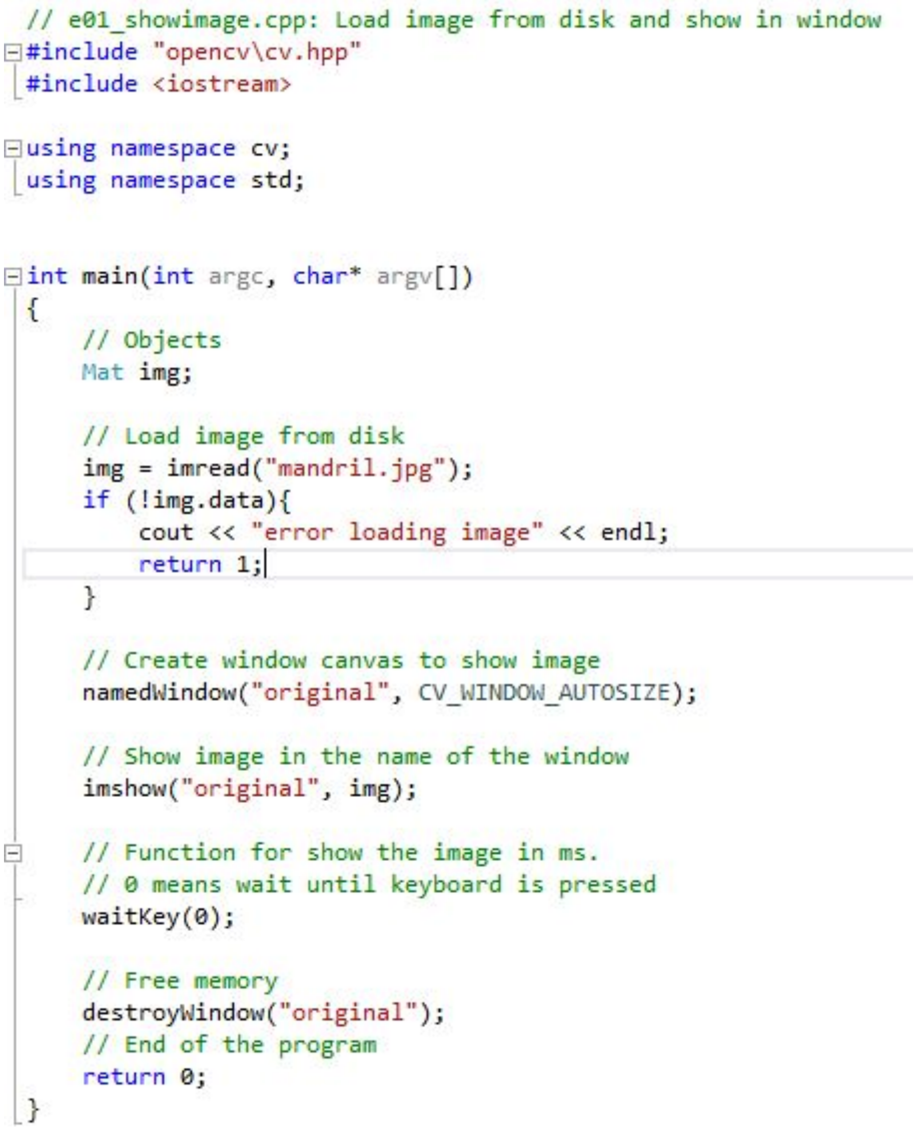
Lesson 1: The OpenCV Libraries

Example 01:

Display an image from disk:

- #include <opencv/cv.hpp>

1. Load the image and check if fail
2. Show the image
3. Wait for any key press
4. Free memory



```
// e01_showimage.cpp: Load image from disk and show in window
#include "opencv\cv.hpp"
#include <iostream>

using namespace cv;
using namespace std;


int main(int argc, char* argv[])
{
    // Objects
    Mat img;

    // Load image from disk
    img = imread("mandril.jpg");
    if (!img.data){
        cout << "error loading image" << endl;
        return 1;
    }

    // Create window canvas to show image
    namedWindow("original", CV_WINDOW_AUTOSIZE);

    // Show image in the name of the window
    imshow("original", img);

    // Function for show the image in ms.
    // 0 means wait until keyboard is pressed
    waitKey(0);

    // Free memory
    destroyWindow("original");
    // End of the program
    return 0;
}
```

Lesson 1: The OpenCV Libraries

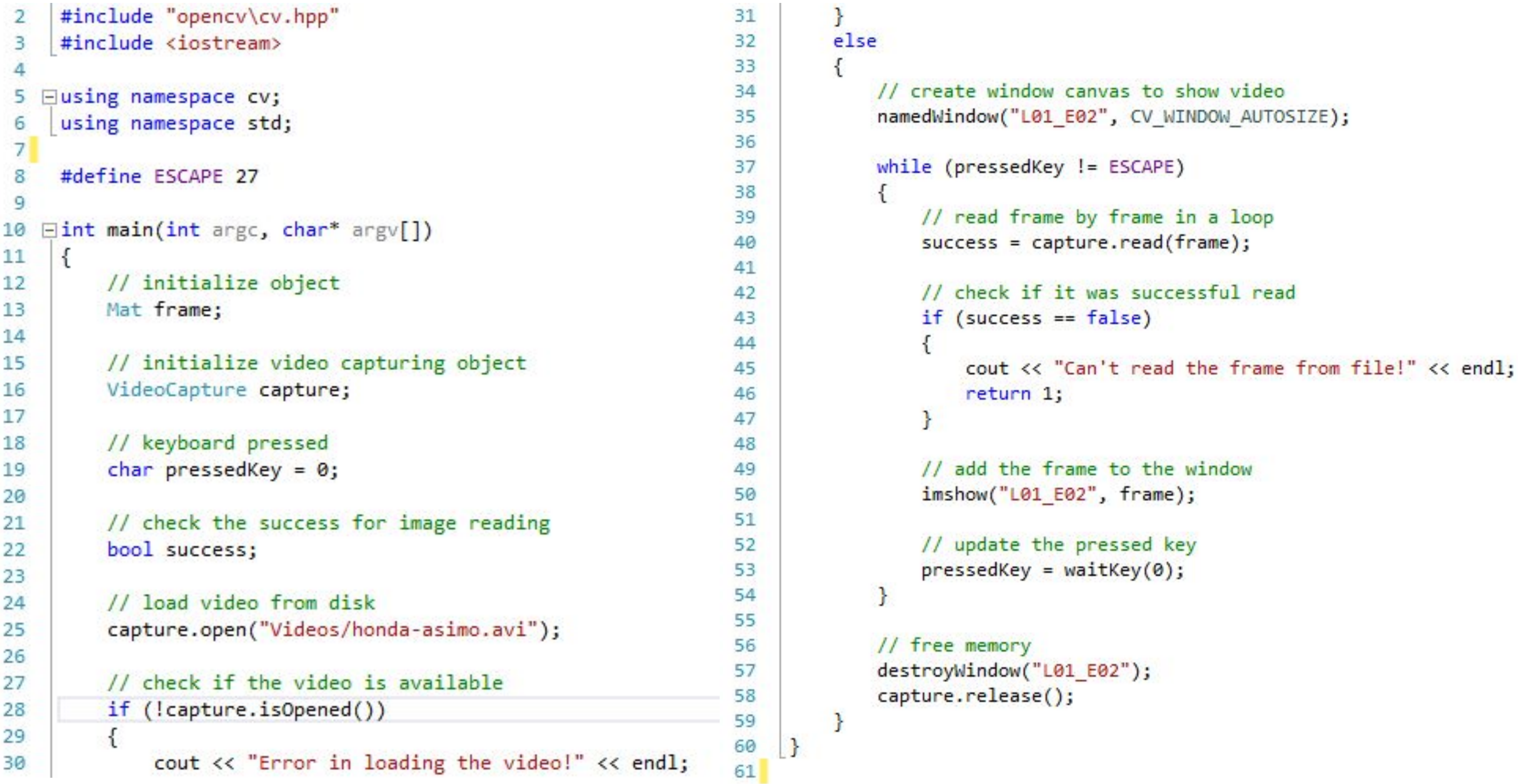
Example 02:

Show image from video:

1. Load the video file using VideoCapture
2. Check for failure
3. Get the first frame
4. Check for failure
5. Show the frame
6. Capture key press
7. If ESCAPE, finish the loop, else show next frame
8. Free memory
9. End program

Lesson 1: The OpenCV Libraries

Example 02: Show image from video



```cpp
#include "opencv\cv.hpp"
#include <iostream>

using namespace cv;
using namespace std;

#define ESCAPE 27

int main(int argc, char* argv[])
{
    // initialize object
    Mat frame;

    // initialize video capturing object
    VideoCapture capture;

    // keyboard pressed
    char pressedKey = 0;

    // check the success for image reading
    bool success;

    // load video from disk
    capture.open("Videos/honda-asimo.avi");

    // check if the video is available
    if (!capture.isOpened())
    {
        cout << "Error in loading the video!" << endl;
    }
    else
    {
        // create window canvas to show video
        namedWindow("L01_E02", CV_WINDOW_AUTOSIZE);

        while (pressedKey != ESCAPE)
        {
            // read frame by frame in a loop
            success = capture.read(frame);

            // check if it was successful read
            if (success == false)
            {
                cout << "Can't read the frame from file!" << endl;
                return 1;
            }

            // add the frame to the window
            imshow("L01_E02", frame);

            // update the pressed key
            pressedKey = waitKey(0);
        }

        // free memory
        destroyWindow("L01_E02");
        capture.release();
    }
}
```

Lesson 1: The OpenCV Libraries

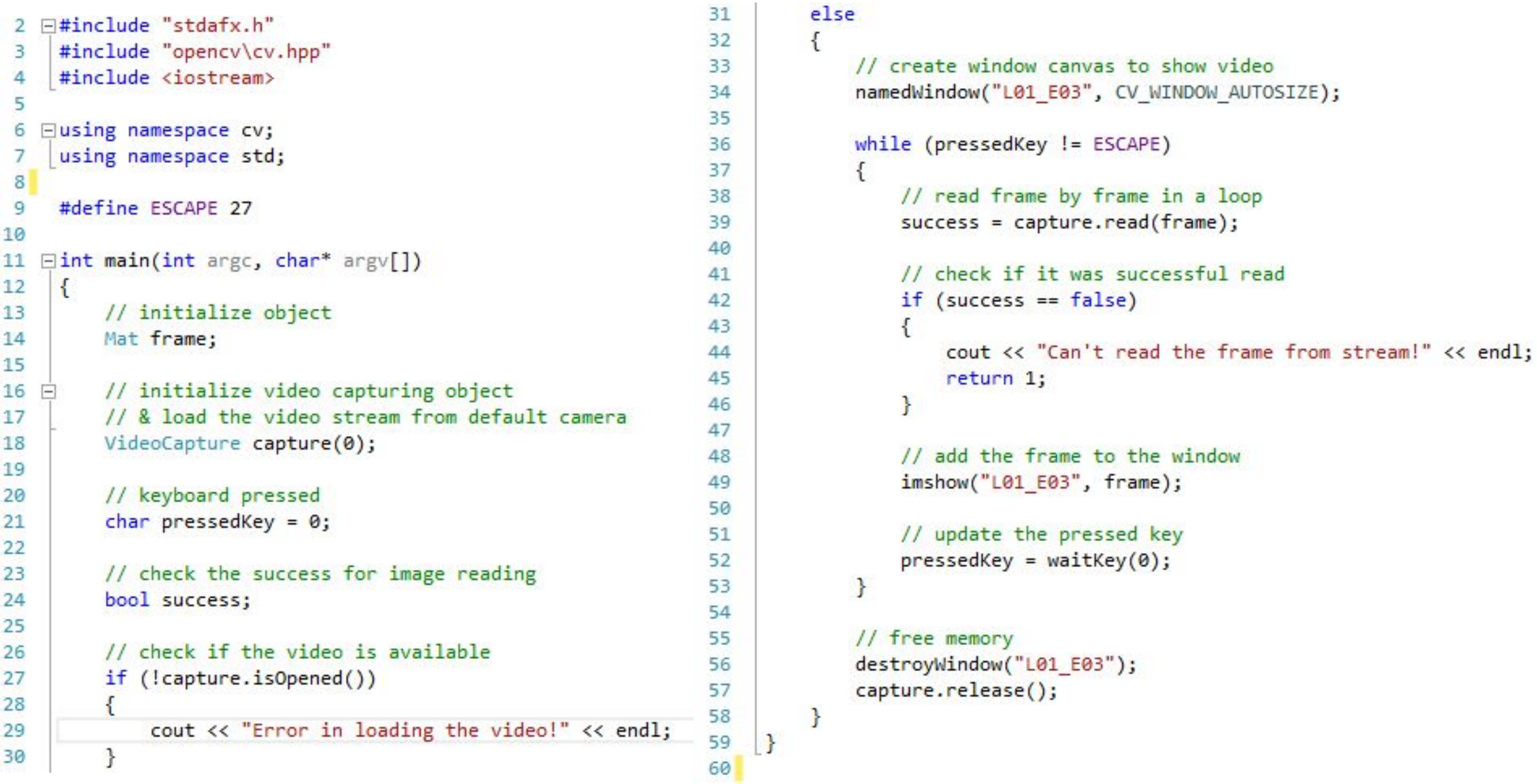
Example 03:

Show image from camera stream:

1. Open camera stream
2. Check for failure
3. Get the first frame
4. Check for failure
5. Show the frame
6. Capture key press
7. If ESCAPE, finish the loop, else show next frame
8. Free memory
9. End program

Lesson 1: The OpenCV Libraries

Example 03: Show image from camera stream



```
#include "stdafx.h"
#include "opencv\cv.hpp"
#include <iostream>

using namespace cv;
using namespace std;

#define ESCAPE 27

int main(int argc, char* argv[])
{
	// initialize object
	Mat frame;

	// initialize video capturing object
	// & load the video stream from default camera
	VideoCapture capture(0);

	// keyboard pressed
	char pressedKey = 0;

	// check the success for image reading
	bool success;

	// check if the video is available
	if (!capture.isOpened())
	{
		cout << "Error in loading the video!" << endl;
	}
	else
	{
		// create window canvas to show video
		namedWindow("L01_E03", CV_WINDOW_AUTOSIZE);

		while (pressedKey != ESCAPE)
		{
			// read frame by frame in a loop
			success = capture.read(frame);

			// check if it was successful read
			if (success == false)
			{
				cout << "Can't read the frame from stream!" << endl;
				return 1;
			}

			// add the frame to the window
			imshow("L01_E03", frame);

			// update the pressed key
			pressedKey = waitKey(0);
		}

		// free memory
		destroyWindow("L01_E03");
		capture.release();
	}
}
```

Lesson 1:
The OpenCV Libraries

Arturo de la Escalera & Jorge Beltrán
Course 2019-2020