Explanation

Title: OpenCV C++ Program to blur a video.
Author: Aditya Prakash
The following is the explanation to the **C++** code to blur a **video** in C++ using the tool **OpenCV**.
Things to know:
(1) The code will only compile in Linux environment.
(2) To run in windows, please use the executable file.
(3) Compile command: g++ -w blur_vid.cpp -o blur_vid `pkg-config --libs opencv`
(4) Run command: ./blur_vid
(5) The video Bumpy.mp4 has to be in the same directory as the code.
Before you run the code, please make sure that you have OpenCV installed on your // system.
Code area:

#include <opencv2/core/core.hpp>
core – Module to define basic data structures, example: Dense Multi-Dimensional array Mat and basic functions too.

#include <opencv2/highgui/highgui.hpp>
highgui - an interface to video and image capturing.

#include <opencv2/imgproc/imgproc.hpp>
imgproc - An image processing module that for linear and non-linear image filtering, geometrical image transformations, color space conversion and so on.

#include <stdio.h>
#include <iostream>
The header files for performing input and output.

using namespace cv;
Namespace where all the C++ OpenCV functionality resides.

using namespace std;
For input output operations.

```
int main()
{
      VideoCapture cap("Bumpy.mp4");
      // cap is the object of class video capture that tries to capture Bumpy.mp4
    if ( !cap.isOpened() )  // isOpened() returns true if capturing has started.
    {
           cout << "Cannot open the video file. \n";
           return -1;
    }

    double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per second.
    // The function get is used to derive a property from the element.
    // Example:
    // CV_CAP_PROP_POS_MSEC :  Current Video capture timestamp.
    // CV_CAP_PROP_POS_FRAMES : Index of the next frame.
```

```
    namedWindow("A_good_name",CV_WINDOW_AUTOSIZE); // Create a window called
      //"A_good_name".
      // first argument: name of the window.
      // second argument: flag- types:
      // WINDOW_NORMAL : The user can resize the window.
      // WINDOW_AUTOSIZE : The window size is automatically adjusted to fitvthe
displayed image() ), and you cannot change the window size manually.
      // WINDOW_OPENGL : The window will be created with OpenGL support.

    While(1) // An infinite loop
    {
            Mat frame;
            // Mat object is a basic image container. frame is an object of Mat.

        if (!cap.read(frame)) // if not success, break loop
        // read() decodes and captures the next frame.
        {
                cout<<"\n Cannot read the video file. \n";
            break;
        }
        blur(frame,frame,Size(10,10)); // To blur the image.
        //  first argument: input source
        //  second argument: output destination
        //  third argument: blurring kernel size



        imshow("A_good_name", frame);
            // first argument: name of the window.
            // second argument: image to be shown(Mat object).

            if(waitKey(30) == 27) // Wait for 'esc' key press to exit
        {
            break;
        }
    }

    return 0;
}

End of explanation.
```