1> Libraries:-

--------------------------------------------------

(a) opencv2/core/core.hpp: It includes all data types like Mat,Point etc.

Mat is used to declare matrices.

Mat img(rows,cols,MACRO,Scalar());

Scalar() function sets the default value of the entries.

Macro specifies the type of entries in the matrix.(eg: CV\_8UC1)

To copy/clone one image into another-

img1=img.clone();

(b) opencv2/highgui/highgui.hpp: It includes all files necessary for reading/displaying images/videos.

(c) using namespace cv:Different classes are included in this namespace.

If this is not used we have to write the code as:

cv::Mat img;

cv::imshow("image",img);

(d) To read image:- imread("loaction of image",Macro);

Macro: CV\_LOAD\_IMAGE\_GRAYSCALE

CV\_LOAD\_IMAGE\_COLOR

(e) To access value at any point/pixel of image:-

img.at<uchar>(i,j) [for 2D matrices]

(f) To create a Window:-

namedWindow("name of window",Macro);

Macro: WINDOW\_AUTOSIZE;

WINDOW\_NORMAL;

(g) To diasplay image:-

imshow("name of window",img);

------------------------------------

TASK1:To create and display image of flag:

------------------------------------

#include "stdafx.h"

#include "opencv2/core/core.hpp"

#include "opencv2/highgui/highgui.hpp"

using namespace cv;

using namespace std;

int main()

{

int i,j;

Mat img(50,500,CV\_8UC1,Scalar(60));

for(i=0;i<50;i++){

for(j=0;j<500;j++){

img.at<uchar>(i,j)=100;

}

}

for(i=50;i<100;i++){

for(j=0;j<500;j++){

img.at<uchar>(i,j)=200;

}

}

for(i=100;i<150;i++){

for(j=0;j<500;j++){

img.at<uchar>(i,j)=255;

}

}

namedWindow("image",WINDOW\_AUTOSIZE);

imshow("image",img);

waitKey(0);

return 0;

}

------------------------------------

TASK2:To read grayscale of a coloured image:

------------------------------------

#include "stdafx.h"

#include "opencv2/core/core.hpp"

#include "opencv2/highgui/highgui.hpp"

using namespace cv;

using namespace std;

int main()

{

Mat img;

img=imread("Image1.jpg",CV\_LOAD\_IMAGE\_GRAYSCALE);

namedWindow("image",WINDOW\_AUTOSIZE);

imshow("image",img);

waitKey(0);

return 0;

}

-----------------------------------