UNIVERSITY OF MINES AND TECHNOLOGY, TARKWA

FIRST SEMESTER EXAMINATIONS, NOV/ DEC 2018

COURSE NO: CE 273
COURSE NAME: COMPUTER GRAPHICS (PRACTICAL)
CLASS: CE II **TIME:** 2 HRS

Name**:** _____Index Number: _____

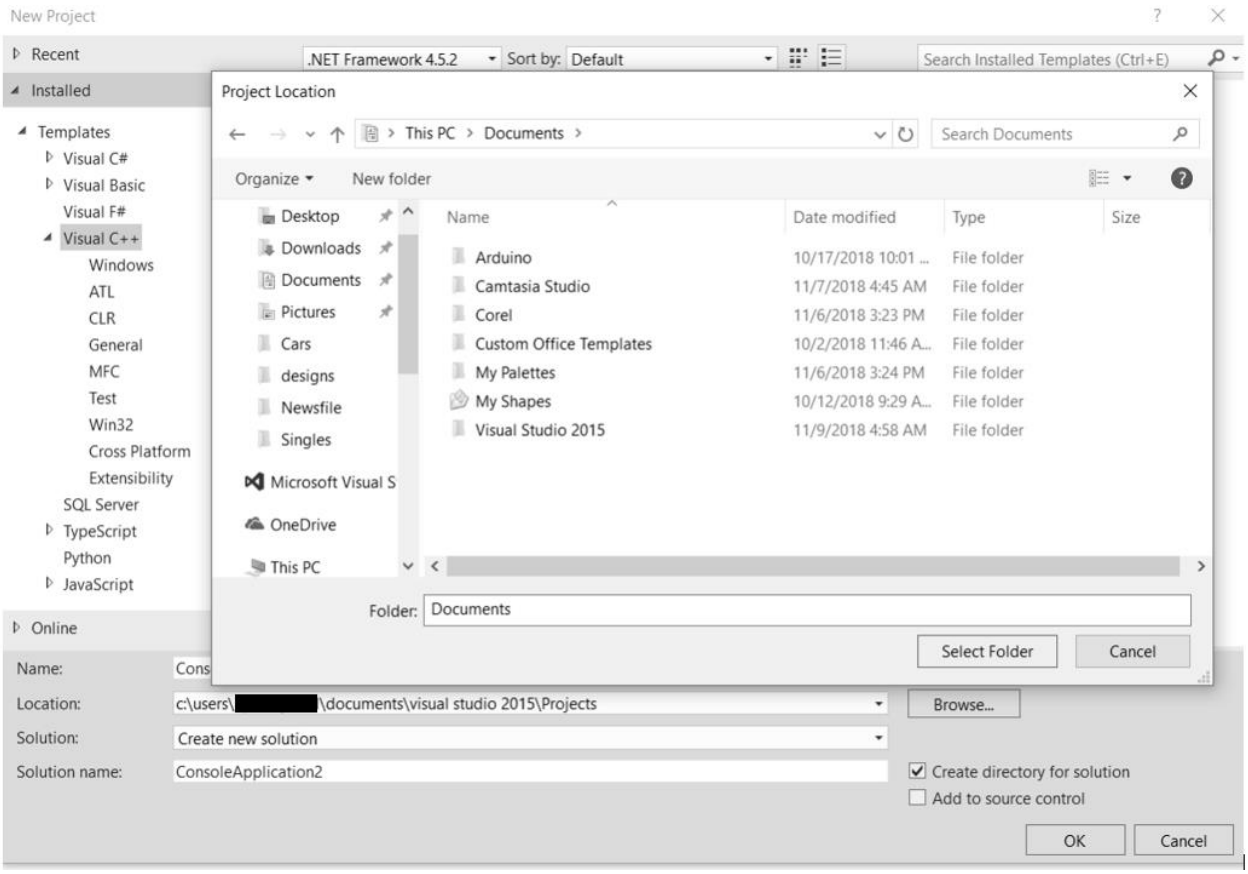
Instructions: *This* ***PRACTICAL SECTION*** *has two ACTIVITIES.* ***Answer ALL QUESTIONS.***

ACTIVITY 1

a. Save the "External Libs" folder (containing GLEW and GLFW libraries) on the Desktop.

b. Start a new project in Visual Studio

c. Select "Documents" as your solution directory (as illustrated below)



d. Add the libraries and include directories for GLEW and GLFW (use the appropriate macros where necessary).

ACTIVITY 2

Using the following template code and keywords, produce the following output using shaders in OpenGL:

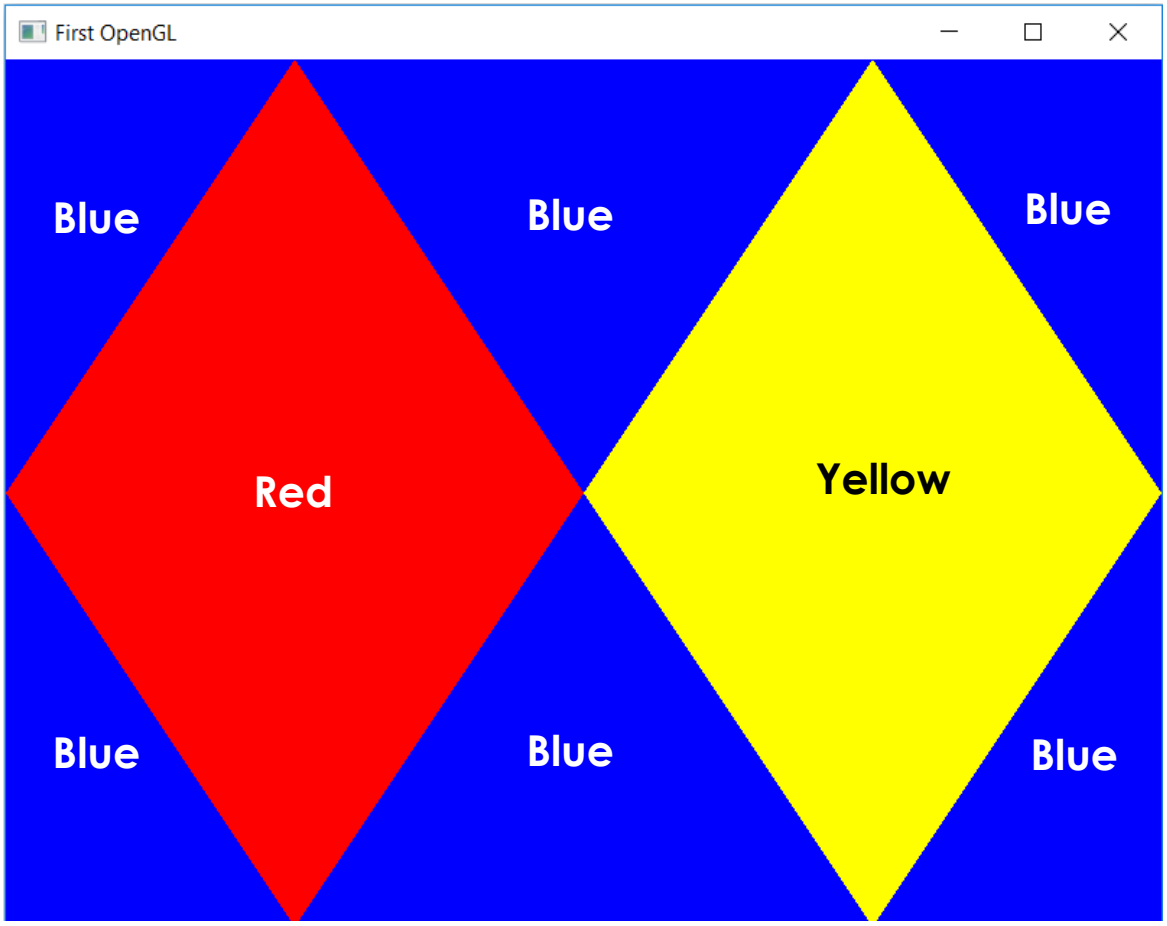
Keywords:
GL_TRIANGLES
GL_VERTEX_SHADER
GL_FRAGMENT SHADER
NB: A combination of red and green colours form yellow colour.

TEMPLATE CODE
NB: This is only a template code, and as such may contain errors

```
const GLint width = 800, height = 600;
GLuint VAO, VBO, shader;

static const char* vertexShader = "                         \n\
#version 330 core                                           \n\
layout (location = 0) in vec3 pos;                          \n\
void main()                                                 \n\
{                                                           \n\
      gl_Position = vec4();                                 \n\
}";

static const char* fragmentShader = "                       \n\
#version 330 core                                           \n\
out vec4 colour;                                            \n\
void main()                                                 \n\
{                                                           \n\
      colour = vec4();                                      \n\
}";


void CreateTriangle()
{
      float positions[] = {
      };

      glGenVertexArrays(1, &VAO);
      glBindVertexArray(VAO);
      glGenBuffers(1, &VBO);
      glBindBuffer(GL_ARRAY_BUFFER, VBO);
      glBufferData(GL_ARRAY_BUFFER, GL_STATIC_DRAW);

      glEnableVertexAttribArray(0);
      glVertexAttribPointer();
}

void AddShader(GLuint theProgram, const char* shaderCode, GLenum shaderType)
{
      GLuint theShader = glCreateShader(shaderType);
```

```
	const GLchar* theCode[1];
	theCode[0] = shaderCode;

	glShaderSource();
	glCompileShader(theShader);
	glAttachShader(theProgram, theShader);
}

void CompileShaders()
{
	shader = glCreateProgram();

	AddShader();
	AddShader();

	glLinkProgram();
	glValidateProgram();
}

int main()
{
	glfwInit();

	GLFWwindow *mainWindow = glfwCreateWindow();
	glfwMakeContextCurrent();
	glewInit();

	int bufferWidth, bufferHeight;
	glfwGetFramebufferSize(mainWindow, &bufferWidth, &bufferHeight);
	glViewport();

	CreateTriangle();
	CompileShaders();
	while (!glfwWindowShouldClose())
	{
		glfwPollEvents();

		glClearColor();
		glClear(GL_COLOR_BUFFER_BIT);

		glUseProgram();
		glBindVertexArray();
		glDrawArrays();

		glfwSwapBuffers();
	}
	glDeleteVertexArrays(1, &VAO);
	glDeleteProgram(shader);
	glfwTerminate();

	return 0;
}
```

FINAL EXPECTED OUTPUT



20 MARKS

Examiners: ***F L Aryeh*** */ R K Annan*