Computer Graphics
– Shaders (Questions)

J.-Prof. Dr. habil. Kai Lawonn

Possible Questions

Correct the vertex and fragment shader:

```
#version 330 core
layout (location = 0) in vec3 aPos;

void main()
{
   gl_Position = vec4(aPos, 1.0);
   vertexColor = vec4(0.5, 0.0, 0.0, 1.0);
}
```

```
#version 330 core
out vec4 FragColor;

void main()
{
   FragColor = vertexColor;
}
```

Possible Questions

Correct the vertex and fragment shader:

```
#version 330 core
layout (location = 0) in vec3 aPos;
out vec4 vertexColor;
void main()
{
   gl_Position = vec4(aPos, 1.0);
   vertexColor = vec4(0.5, 0.0, 0.0, 1.0);
}
```

```
#version 330 core
out vec4 FragColor;
in vec4 vertexColor;
void main()
{
   FragColor = vertexColor;
}
```

Possible Questions

Complete the C++ source code and fragment shader:

```
glUseProgram(shaderProgram);

float greenValue = 0.5f;

int vertexColorLocation = glGetUniformLocation(shaderProgram, □);
glUniform4f(vertexColorLocation, 0.0f, greenValue, 0.0f, 1.0f);
```

```
#version 330 core
out vec4 FragColor;
□ vec4 ourColor;
void main()
{
   FragColor = ourColor;
}
```

Possible Questions

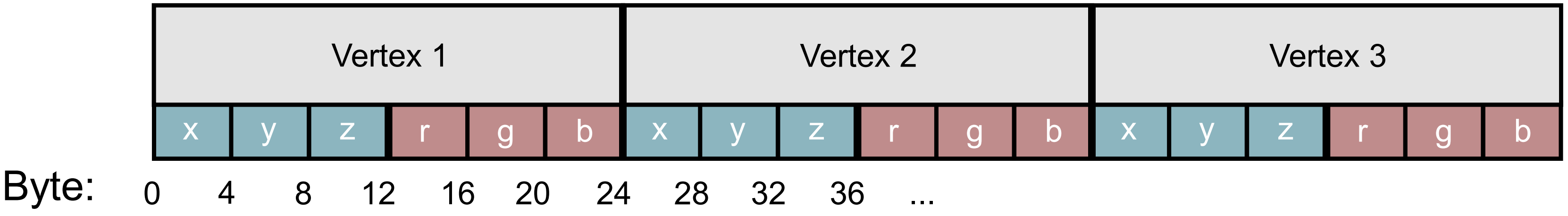
Complete the C++ source code and fragment shader:

```
glUseProgram(shaderProgram);

float greenValue = 0.5f;

int vertexColorLocation = glGetUniformLocation(shaderProgram, "ourColor");
glUniform4f(vertexColorLocation, 0.0f, greenValue, 0.0f, 1.0f);
```

```
#version 330 core
out vec4 FragColor;
uniform vec4 ourColor;
void main()
{
   FragColor = ourColor;
}
```

Possible Questions

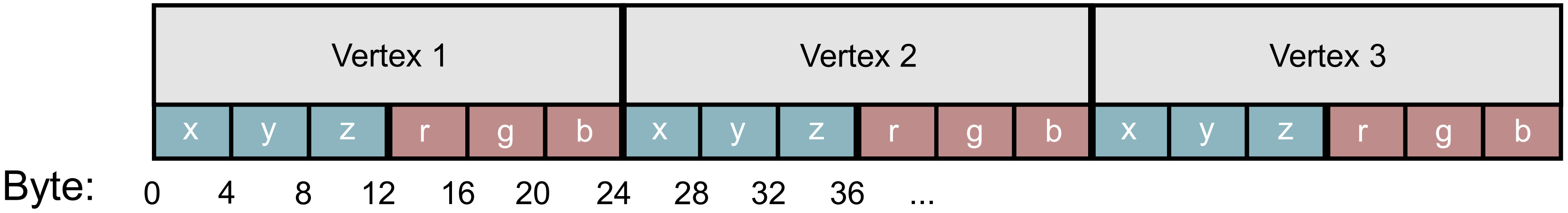
Fill in the correct stride and offset (float values):



```
// position attribute
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, ____________________, (void*)(____________________));
glEnableVertexAttribArray(0);
// color attribute
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, ____________________, (void*)(____________________));
glEnableVertexAttribArray(1);
```

Possible Questions

Fill in the correct stride and offset (float values):



Vertex 1 | Vertex 2 | Vertex 3

x y z r g b | x y z r g b | x y z r g b

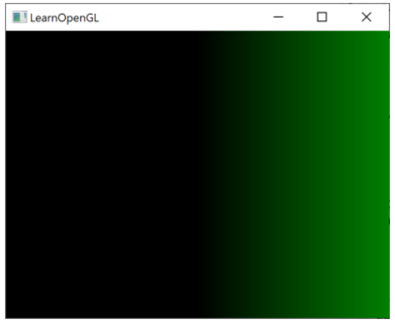
Byte: 0 4 8 12 16 20 24 28 32 36 …

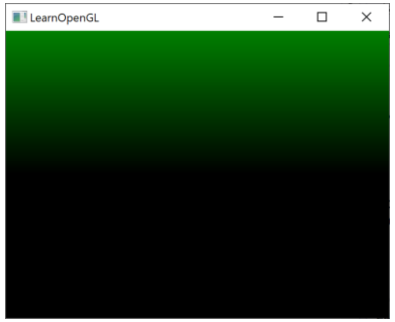
```
// position attribute
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(float), (void*)0);
glEnableVertexAttribArray(0);
// color attribute
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(float), (void*)(3 * sizeof(float)));
glEnableVertexAttribArray(1);
```

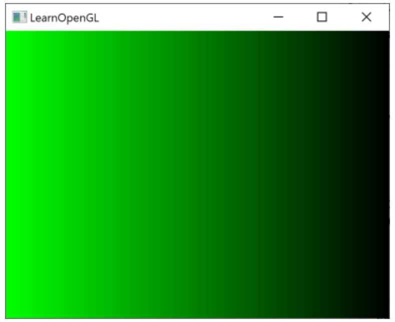
Possible Questions

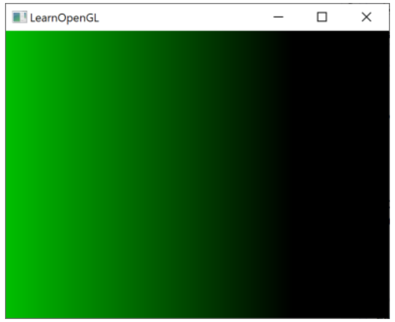
What is the correct output (window 800x600)?

```
#version 330 core
out vec4 FragColor;

void main()
{
  vec2 p = gl_FragCoord.xy/vec2(800,600)-vec2(0.5);
  if(p.x<0.5)
    FragColor = vec4 (0,p.x,0,1);
  else
    FragColor = vec4 (1,0,0,1);
}
```





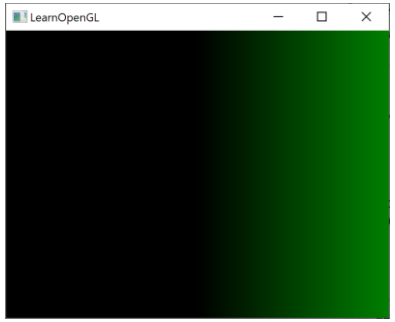


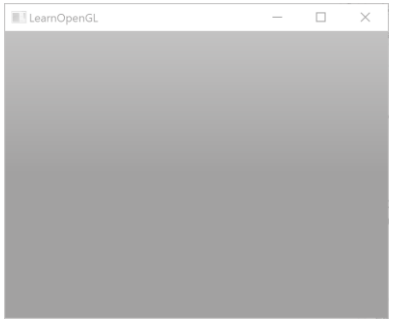


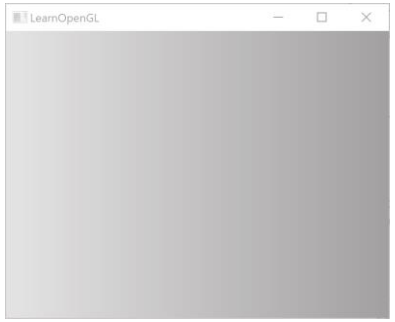
Possible Questions

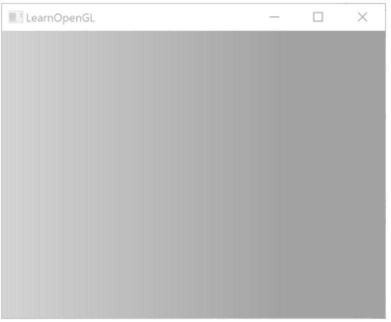
What is the correct output (window 800x600)?

```
#version 330 core
out vec4 FragColor;

void main()
{
  vec2 p = gl_FragCoord.xy/vec2(800,600)-vec2(0.5);
  if(p.x<0.5)
    FragColor = vec4 (0,p.x,0,1);
  else
    FragColor = vec4 (1,0,0,1);
}
```





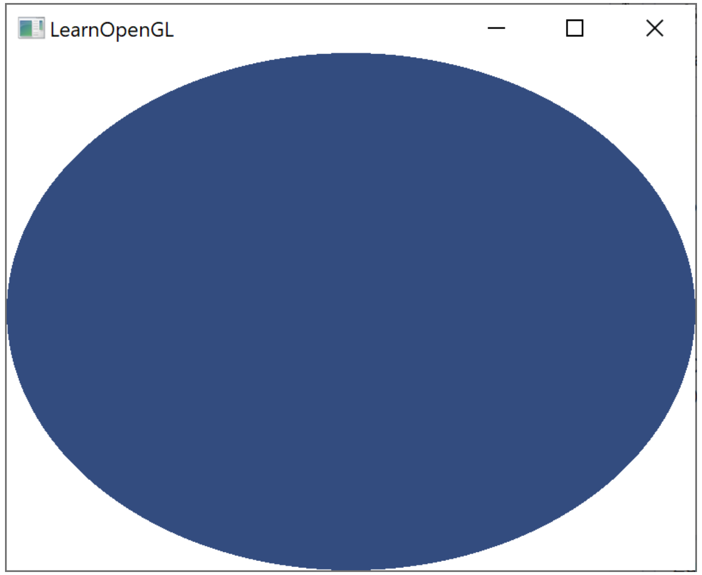




Possible Questions

Complete the code to get the following output:

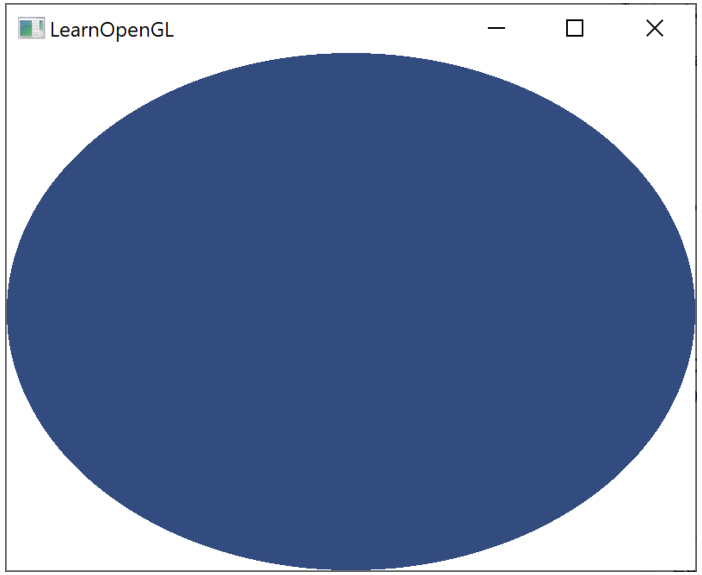
```
#version 330 core
out vec4 FragColor;

void main()
{
  vec2 p = gl_FragCoord.xy/vec2(800,600)-vec2(0.5);
  p=p*2; //p in [-1,1]
  if(_________________)
    FragColor = vec4(0.2,0.3,0.5,1);
  else
    FragColor = vec4(1);
}
```



Possible Questions

Complete the code to get the following output:

```
#version 330 core
out vec4 FragColor;

void main()
{
  vec2 p = gl_FragCoord.xy/vec2(800,600)-vec2(0.5);
  p=p*2; //p in [-1,1]
  if(p.x*p.x+p.y*p.y<1)
    FragColor = vec4(0.2,0.3,0.5,1);
  else
    FragColor = vec4(1);
}
```


LearnOpenGL