

5. Ability to render a cube with predefined vertex information.

Problem::Answer::

```
std::vector<Vertex> RenderingGeometryApp::genCube(std::vector<Vertex> vertices)
{
    std::vector<Vertex> CubePoints;

    CubePoints.push_back(Vertex(glm::vec4(0, 1, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 1, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 0, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 0, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 0, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 0, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 1, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 1, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 1, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 1, 1, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 1, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(1, 0, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 1, 0, 1), glm::vec4(1)));
    CubePoints.push_back(Vertex(glm::vec4(0, 0, 0, 1), glm::vec4(1)));
    return CubePoints;
}
```