**Study Table**

**Weeks 1-2: Foundations of Computer Graphics**

* Read chapters 1-3 of "Computer Graphics: Principles and Practice" for an introduction to the field, coordinate systems, transformations, and basic rendering concepts.
* Simultaneously, go through relevant sections in "OpenGL Programming Guide" to start understanding OpenGL basics.

**Weeks 3-4: Raster Graphics and Geometric Modeling**

* Dive into chapters 4-6 of "Computer Graphics: Principles and Practice" covering raster graphics, scan conversion, and geometric modeling.
* Begin implementing basic shapes and transformations using OpenGL based on your understanding from the OpenGL Programming Guide.

**Weeks 5-6: 3D Viewing, Shading, and Illumination**

* Study chapters 7 and 8 of "Computer Graphics: Principles and Practice" focusing on 3D viewing and shading models.
* Experiment with OpenGL shading techniques, working through examples from "OpenGL Programming Guide."

**Weeks 7-8: Advanced Rendering Techniques**

* Read chapters 9-11 of "Computer Graphics: Principles and Practice" covering color models, texture mapping, and introductory ray tracing concepts.
* Start implementing ray tracing concepts using C++ and OpenGL, possibly referring to resources from "OpenGL Programming Guide" and online tutorials.

**Weeks 9-10: Animation and Real-Time Rendering**

* Study the animation principles in "Computer Graphics: Principles and Practice" (chapters 12-13).
* Begin exploring "Real-Time Rendering" to understand real-time graphics concepts and techniques.

**Weeks 11-12: Specialized Rendering Techniques**

* Study more advanced rendering techniques from "Physically Based Rendering" to gain insights into realistic rendering.
* Continue your real-time graphics exploration using resources from "Real-Time Rendering."

**Weeks 13-14: Graphics Math and Integration**

* Dedicate time to understanding the mathematical foundations of computer graphics. Study chapters from your "Computer Graphic Math" resource.
* Work on integrating your knowledge by tackling small projects that involve both mathematical concepts and rendering techniques.

**Weeks 15-16: Project Work and Recap**

* Apply your accumulated knowledge to a larger project. This could involve creating a simple 3D scene, implementing a basic rendering pipeline, or experimenting with ray tracing techniques.
* Review and recap key concepts from all the resources you've studied. Solidify your understanding through practice problems and coding challenges.