FORMAN CHRISTIAN COLLEGE (A CHARTERED UNIVERSITY), LAHORE
DEPARTMENT OF COMPUTER SCIENCE

COURSE OUTLINE

CS 453 — Computer Graphics

BS Computer Science | NQF Level 6 | Fall Semester 2026

Aligned with HEC Undergraduate Education Policy V 1.1 (2023)

1. Course Information

Course Code	CS 453	**Course Title**	Computer Graphics
Credit Hours	3 (3–0)	**Contact Hours**	3 theory / week (MWF, 50 min)
Course Type	Domain Elective (CS)	**Semester**	Fall 2026 (Semester 7)
Pre-requisites	Data Structures; basic probability; working proficiency in Python	**Co-requisites**	None
Programme	BS Computer Science	**Knowledge Area**	Graphics & Visualization
Language / Tool	Python 3.x, NumPy, Pillow; GLSL (ModernGL)	**Semester Weeks**	15 weeks (41 lectures)
Instructor	Fakhir Shaheen	**Email**	fakhirshaheen@fccollege.edu.pk
Office Hours	Posted on course website	**Classroom**	*As per timetable*

Note: There is no separately scheduled lab; the practical component is carried by the weekly programming assignments and the independent final project.

2. Course Introduction / Description

This senior-level elective teaches how images are computed from scene descriptions — the "scene → *your code* → image" idea. The organising principle of the course is that students do not *use* graphics tools; they *build* them. In the offline half, every renderer is written from scratch in Python + NumPy: raster images and color, line and curve drawing, transformation hierarchies and scene graphs, and a complete recursive ray tracer with diffuse, reflective and refractive materials, antialiasing, and triangle meshes — culminating in the rendering equation and a Monte Carlo path tracer. The real-time half moves to the GPU: the modern programmable pipeline, vertex and fragment shaders in GLSL, procedural (SDF) shading, perspective viewing, real-time lighting, texturing and shadow mapping, closing with a survey of the field's frontier — neural and differentiable rendering.

Mathematics (vectors, matrices, homogeneous and barycentric coordinates) is introduced *just-in-time*, at the moment a project needs it. Every week produces a visible rendered artifact: eleven weekly assignments (A1–A11) culminate in an independent final project submitted with a recorded demonstration. The course prepares students for work or graduate study in games, film and visual effects, scientific visualization, simulation, and the rapidly growing intersection of graphics and machine learning.

3. Programme Learning Outcomes (PLOs)

The PLOs below are adopted from the HEC Revised Curriculum for Computer Science (2025) and the Seoul Accord graduate attributes.

PLO	Title	Statement
PLO-1	Academic Education	Prepare graduates as computing professionals.
PLO-2	Knowledge for Solving Computing Problems	Apply computing, mathematics and domain knowledge to abstract and conceptualise computing models from defined problems.
PLO-3	Problem Analysis	Identify, formulate and solve complex computing problems, reaching substantiated conclusions from first principles.
PLO-4	Design / Development of Solutions	Design and evaluate solutions, systems, components or processes meeting specified needs, with due regard for health, safety, cultural, societal and environmental factors.
PLO-5	Modern Tool Usage	Create, select and apply appropriate techniques, resources and modern computing tools, understanding their limitations.
PLO-6	Individual and Team Work	Function effectively as an individual and as a member or leader of diverse, multi-disciplinary teams.
PLO-7	Communication	Communicate effectively with the computing community and society at large through reports, documentation, presentations and clear instructions.
PLO-8	Computing Professionalism and Society	Assess societal, health, safety, legal and cultural issues, and the responsibilities these place on professional computing practice.
PLO-9	Ethics	Commit to professional ethics, responsibilities and norms of computing practice.
PLO-10	Life-long Learning	Recognise the need, and have the ability, to engage in independent, continual professional learning.

4. Course Learning Outcomes (CLOs)

By the end of this course the student will be able to:

CLO No.	Course Learning Outcome	Bloom's Taxonomy	PLO
CLO-1	**Explain** how images are formed from scene descriptions: raster images and color, the ray-tracing and rasterization approaches, sampling and antialiasing, and the architecture of the modern GPU pipeline — including the cost/quality trade-offs between offline and real-time rendering.	BT2 (Understand)	PLO-2
CLO-2	**Apply** vectors, matrices, parametric curves, and homogeneous and barycentric coordinates to model, animate, and hierarchically organise 2D and 3D scenes (reference frames, scene graphs).	BT3 (Apply)	PLO-3
CLO-3	**Construct** a recursive ray tracer from first principles: cameras and viewing rays, ray–object intersection, diffuse and specular shading, shadows, reflection, refraction, and ray-traced triangle meshes.	BT6 (Create)	PLO-4
CLO-4	**Develop** real-time interactive graphics on the GPU using modern OpenGL: vertex and fragment shaders in GLSL, the model–view–projection chain, real-time lighting, texture mapping, and procedural (SDF) shading.	BT3 (Apply)	PLO-5
CLO-5	**Propose, scope and deliver** an independent graphics project, communicating design decisions and results through a written proposal, an in-class pitch, milestone check-ins, and a recorded demonstration of the final deliverable.	BT6 (Create)	PLO-7

CLO–PLO Articulation (Mapping) Matrix

Strength of contribution: **H** = High (directly taught and directly assessed), M = Medium (taught and partially assessed), L = Low (introduced or reinforced), – = not addressed at this level.

CLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10
CLO-1	L	**H**	M	–	–	–	–	–	–	L
CLO-2	L	M	**H**	M	L	–	–	–	–	L
CLO-3	L	M	M	**H**	M	–	–	–	–	L
CLO-4	L	L	L	M	**H**	–	–	–	–	M
CLO-5	L	L	M	M	M	–	**H**	–	L	M

Mapping rationale. **PLO-2** ← CLO-1 (image-formation concepts and models); **PLO-3** ← CLO-2 (mathematical formulation of scenes from first principles); **PLO-4** ← CLO-3 (designing and evaluating a complete renderer); **PLO-5** ← CLO-4 (GPU, OpenGL/GLSL toolchain and its limitations); **PLO-7** ← CLO-5 (proposal, check-ins, live demonstration). **PLO-1** is programme-level, hence Low throughout. **PLO-6** is not assessed: all graded work is individual (peer critique is formative only). **PLO-8** belongs to other courses. **PLO-9** appears at Low via academic integrity; **PLO-10** at Medium — the closing survey of neural rendering demands independent engagement with a rapidly evolving field.

5. Course Outline (Contents)

Raster images, pixels and color; alpha blending; line rasterisation (Bresenham); region filling (analytic inside-tests, flood fill); vectors, the dot and cross products; parametric lines, circles and ellipses; Bézier curves and de Casteljau's algorithm; 2×2 and 3×3 transformation matrices; homogeneous coordinates; composition and order of transforms; reference frames and the scene graph; rays and ray–object intersection; 2D raycast lighting and shadows; 3D cameras and viewing rays; ray–sphere and ray–triangle intersection; barycentric coordinates; diffuse (Lambert) and specular shading; shadow rays; recursive reflection and refraction (Snell's law); antialiasing; bounding volume hierarchies and ray–box intersection; distribution ray tracing *(stratified sampling, soft shadows, depth of field)*; triangle meshes and the OBJ format; light and radiometry *(flux, irradiance, radiance, solid angle)*; the rendering equation and BRDFs; Monte Carlo path tracing and importance sampling; the real-time loop; GPU architecture and the programmable pipeline; GLSL vertex and fragment shaders, attributes and uniforms; signed distance fields and raymarching; the model–view–projection chain and perspective projection; Blinn–Phong lighting; texture and UV mapping; shadow mapping; neural rendering: NeRFs and Gaussian splatting *(survey)*; independent final project.

Coverage note: Follows the HEC domain-elective specification for Computer Graphics and the ACM/IEEE Graphics & Visualization knowledge area, extended with a survey of neural and differentiable rendering.

6. Weekly Lecture Plan

Wk	Lec	Date	Topic	Deliverable	CLO
1	1	Mon, 31 Aug	What graphics is (and isn't); setup; first pixels & gradients	—	CLO-1
1	2	Wed, 2 Sep	Bresenham's line algorithm	—	CLO-1
1	3	Fri, 4 Sep	Color & alpha; primitives; fills (inside-test, flood fill)	A1 released	CLO-1
2	4	Mon, 7 Sep	Vectors & the dot product	—	CLO-2
2	5	Wed, 9 Sep	Parametric lines, circles, ellipses	—	CLO-2
2	6	Fri, 11 Sep	Bézier curves; de Casteljau	A1 due; A2 released	CLO-2
3	7	Mon, 14 Sep	2×2 matrices: rotation, scale	—	CLO-2
3	8	Wed, 16 Sep	Homogeneous coordinates; composition & order	—	CLO-2
3	9	Fri, 18 Sep	Reference frames & the scene graph	A2 due; A3 released	CLO-2
4	10	Mon, 21 Sep	Rays; marching & nearest hit	—	CLO-2, 3
4	11	Wed, 23 Sep	2D light & shadow: fan of rays, occluders	—	CLO-3
4	12	Fri, 25 Sep	Exact ray–segment intersection (2×2 solve); soft shadows from area lights in 2D	A3 due; A4 released	CLO-2, 3
5	13	Mon, 28 Sep	3D vectors; cross product; the camera	—	CLO-2, 3
5	14	Wed, 30 Sep	Ray–sphere intersection	—	CLO-3
5	15	Fri, 2 Oct	Multi-object scenes; nearest hit; antialiasing	A4 due; A5 released	CLO-3
6	16	Mon, 5 Oct	Normals; diffuse (Lambert) shading; point lights	—	CLO-3
6	17	Wed, 7 Oct	Shadow rays; reflection & metal	—	CLO-3
6	18	Fri, 9 Oct	Refraction & glass (Snell); recursive rays	A5 due; A6 released	CLO-3

Continued on next page...

Wk	Lec	Date	Topic	Deliverable	CLO
7	19	Mon, 12 Oct	Why ray tracing is slow; acceleration ideas	—	CLO-1, 3
7	20	Wed, 14 Oct	Building a BVH: AABBs, the slab test, construction & traversal	—	CLO-3
7	21	Fri, 16 Oct	Distribution ray tracing: stratified sampling, soft shadows, depth of field	A6 due; A7 released	CLO-3
8	22	Mon, 19 Oct	Render show-and-tell; midterm review	—	CLO-1–3
8	—	Wed, 21 Oct	**MIDTERM EXAMINATION**	—	CLO-1–3
8	—	Fri, 23 Oct	*Fall Break — no class*	A7 due (midterm scene)	—
9	23	Mon, 26 Oct	Ray–triangle intersection; barycentric coordinates	—	CLO-2, 3
9	24	Wed, 28 Oct	Meshes & the OBJ format; loading real models	—	CLO-3
9	25	Fri, 30 Oct	Light & radiometry: flux, irradiance, radiance, solid angle	A8 released	CLO-1, 3
10	26	Mon, 2 Nov	The rendering equation & BRDFs	—	CLO-1, 3
10	27	Wed, 4 Nov	Monte Carlo integration; path tracing, emissive materials	—	CLO-3
10	28	Fri, 6 Nov	Importance sampling; the path tracer in practice	A8 due; A9 released	CLO-3
11	—	Mon, 9 Nov	*Iqbal Day — no class*	—	—
11	29	Wed, 11 Nov	Real-time: input $\to$ update $\to$ draw, every frame	—	CLO-1
11	30	Fri, 13 Nov	The GPU; the programmable pipeline; the depth buffer	A9 due; A10 released	CLO-1
12	31	Mon, 16 Nov	Hello triangle (ModernGL, GLSL)	—	CLO-4
12	32	Wed, 18 Nov	Fragment shaders: color $= f(x, y)$ per pixel	—	CLO-4
12	33	Fri, 20 Nov	Shadertoy: SDFs & raymarching	—	CLO-4
13	34	Mon, 23 Nov	3D on the GPU: the MVP chain; perspective	—	CLO-2, 4
13	35	Wed, 25 Nov	Blinn–Phong lighting; textures & UV mapping	—	CLO-4
13	36	Fri, 27 Nov	Final-project kickoff; idea pitches	A10 due; A11 released; Proposal assigned	CLO-5
14	37	Mon, 30 Nov	Project build session: scoping check-ins	Proposal due	CLO-5
14	38	Wed, 2 Dec	Build session: interactive 3D scene	—	CLO-4
14	39	Fri, 4 Dec	Shadow mapping: the depth buffer, two-pass rendering, PCF	A11 due	CLO-4
15	40	Mon, 7 Dec	Graphics meets AI: NeRFs, Gaussian splatting	—	CLO-1
15	41	Wed, 9 Dec	Course review & exam preparation	Final project due after the exam period	CLO-1–4
15	—	Thu, 10 Dec	**FINAL EXAMINATION**	—	CLO-1–4

7. Teaching and Learning Methodology

- **Build, don't use:** every rendering tool in the course — rasteriser, ray tracer, shader pipeline — is implemented by the students themselves; commercial tools (Blender, Photoshop) are deliberately absent.
- **Interactive lectures with live coding:** each technique is written, run, and deliberately broken in front of the class before students apply it.
- **Math just-in-time:** vectors, matrices, homogeneous and barycentric coordinates arrive at the moment a project needs them, not as an up-front block.
- **Weekly visible deliverables (A1–A11):** every week ends in a rendered artifact of the student's

own design — a scene, an animation, a ray-traced render, a live shader — building a personal portfolio across the semester.

- **Offline first, then real time:** students build a full software ray tracer before meeting the GPU, so the pipeline's abstractions (interpolation, shading, transforms) are things they have already implemented by hand.
- **Two clean halves:** the offline half closes at its theoretical peak — radiometry, the rendering equation, Monte Carlo path tracing, importance sampling — before the course switches to the GPU; the hardest early ideas (sampling, shadows, reflection) return at senior depth in each half (BVHs and distribution ray tracing offline, shadow mapping in real time).
- **Independent final project**, run largely outside class time: a written proposal, an in-class pitch, milestone check-ins, and a final submission with a recorded video demonstration.

8. Assessment Scheme and CLO Mapping

Assessment Instrument	Count	Weight (%)	CLOs Assessed
Weekly Programming Assignments (A1–A11)	11	40	CLO-2, CLO-3, CLO-4
Final Project *(proposal, check-ins, video demo)*	1	20	CLO-4, CLO-5
Midterm Examination	1	15	CLO-1, CLO-2, CLO-3
Final Examination *(comprehensive)*	1	25	CLO-1–CLO-4
Total		**100**	

CLO–Assessment Coverage

CLO	Assignments	Project	Midterm	Final
CLO-1	–	–	✓	✓
CLO-2	✓	–	✓	✓
CLO-3	✓	–	✓	✓
CLO-4	✓	✓	–	✓
CLO-5	–	✓	–	–

Every CLO is assessed by at least two independent instruments — CLO-5 through the project's three graded components (proposal, milestone check-ins, final submission with video demonstration) — and CLO-1 through CLO-4 all appear in the final examination, so that attainment can be computed per CLO for OBE reporting.

9. Grading Policy

Grading is **relative** (norm-referenced): final letter grades are awarded by ranking students' aggregate scores against the section's score distribution, in accordance with the university's relative grading policy, rather than by fixed percentage cutoffs. Assessment instruments are nevertheless blueprinted against the CLOs and moderated before use, so that the ranking reflects attainment of the course outcomes.

For Outcome-Based Education reporting, CLO attainment is computed from *raw (uncurved) item scores*: a CLO is deemed *attained* when at least 60% of the enrolled students score 50% or above on the items mapped to that CLO; CLOs falling below this threshold are recorded in the course file with a corrective action for the next offering.

Per HEC policy, a student acquiring less than 2.00/4.00 GPA in a semester but passing all papers is promoted on probation, subject to achieving above 2.00 GPA in the following semester.

10. Reference Materials

Text Book

1. Steve Marschner and Peter Shirley, *Fundamentals of Computer Graphics*, 5th ed., CRC Press, 2021.

Reference Books (or any other standard and latest books)

2. Gabriel Gambetta, *Computer Graphics from Scratch: A Programmer's Introduction to 3D Rendering*, No Starch Press, 2021 *(free online edition available)*.
3. Peter Shirley, Trevor D. Black and Steve Hollasch, *Ray Tracing in One Weekend* series, 2020–2025 *(free online)*.
4. Lee Stemkoski and Michael Pascale, *Developing Graphics Frameworks with Python and OpenGL*, CRC Press, 2021 *(open access)*.

Online Resources

5. *Computer Graphics from Scratch* (free edition) — `https://gabrielgambetta.com/computer-graphics-from-scratch/`
6. *Ray Tracing in One Weekend* — `https://raytracing.github.io/`
7. LearnOpenGL — `https://learnopengl.com/`
8. The Book of Shaders — `https://thebookofshaders.com/`
9. Shadertoy — `https://www.shadertoy.com/`
10. ModernGL documentation — `https://moderngl.readthedocs.io/`

11. Course Policies

Attendance.
There is no formal attendance requirement. Students remain responsible for all material and announcements; the midterm and the final examination require presence in person.

Academic Integrity.
All graded work is individual. Discussion of ideas and techniques is encouraged; submission of another person's code or renders — from a classmate, a repository, or a generative AI tool — as one's own is plagiarism and is reported under the university's disciplinary policy. Students must be able to explain, and modify on request, any line of code they submit.

Use of AI Assistants.
Permitted for explanation and for debugging code the student has already written; not permitted for generating solutions to assessed work. Any AI use must be disclosed in the submission header.

Late Submission.
Assignment deadlines are **hard**: **no late submissions** are accepted, and a missed deadline scores zero. The same applies to the project proposal and final-project deadlines. Exceptions are made only on documented medical or university grounds.

Communication.
Course announcements, slides and assignment specifications are distributed through the course website.