Pokhara University
Faculty of Science and Technology

Course Code: CMP 234	Full Marks: 100
Course title: Computer Graphics (3-1-2)	Pass Marks: 45
Nature of the Course: Theory and Practical	Total Lectures: 45 hours
Level: Bachelor	Program: BE

1. Course Description

This course intends to provide students with an in-depth knowledge of a machine's perception of visual information for storing, accessing and manipulating visual representations of graphical objects and scenes. The course contents cover theoretical, practical, mathematical and algorithmic aspects of concepts such as graphical hardware, two dimensional and three-dimensional representations of graphical objects, the use of techniques such as projection, visible surface detection and lightning effects. The main aim of the course is also to impart knowledge to students regarding the techniques used for incorporating visual realism in the computer-generated images and scenes portrayed on a display device.

2. General Objectives

The course is designed with the following objectives:

- To make the students familiar with the technologies used for representing and rendering of graphical objects and scenes in a computer system
- To equip students with necessary skills and knowledge of Two dimensional and Three dimensional graphics necessary for viewing and performing transformations
- To provide students with a good amount of knowledge of algorithms and approaches used to incorporate maximum amount of visual realism in the computer generated graphical scenes and objects.
- To familiarize students with the graphical standards used for building platform independent portable software

3. Methods of Instruction

As the course consists of theoretical, mathematical and practical aspects of Computer Graphics, the delivery needs to include lecture, tutorial, practical classes. Apart from that, in order to share knowledge and assimilate new ideas and emerging trends, students can be involved in group discussions and presentations pertaining to the field of Computer Graphics. Short quizzes can be held in the class to check the students' level of comprehension. Project work can be assigned to students to come up with a graphical software product that demonstrates their level of understanding of the contents studied in the course.

4. Contents in detail with specific objectives

Specific Objectives	Contents
Provides a brief introduction of the field of Computer Graphics and familiarize students with basic graphics hardware and software. It intends to enhance the understanding of the technology used for producing graphical output on display devices.	**Unit 1: Overview of Computer Graphics and Graphics Systems (7 hrs)** 1.1 Introduction, Applications and Recent trends of Computer Graphics 1.2 Interactive Input Devices 1.3 Display Devices, Color Monitors, Hard Copy Devices 1.4 Raster and Random Scan Systems and Architectures 1.5 Video Controller 1.6 Use of Digital to Analog Converter and Frame Buffer

	Organization 1.7 Graphics Software, Modern Graphics Hardware (GPU)
Learn about various scan conversion and area filling algorithms. Students will be able to implement these scan conversion algorithms using a programming language. This chapter helps students learn how algorithms are used to generate output primitives and fill them with specified intensities.	**Unit 2: Scan Conversion Algorithms (7 hrs)** 2.1 Scan Conversion 2.2 Line Drawing Algorithms 2.2.1 Digital Differential Analyzer 2.3.2 Bresenham's Algorithm 2.3 Circle Generation Algorithm 2.4 Ellipse Generation Algorithm 2.5 Filled Area Primitives 2.5.1 Scan Line Polygon Fill Algorithm 2.5.2 Boundary Fill Algorithm, Flood Fill Algorithm
Get students well acquainted with the two-dimensional geometric transformations and viewing Able to make use of various transformation operations and viewing operations to reposition and resize objects in two dimensional scene Learn about the operations used in viewing routines that convert a world coordinate scene description to a display for an output device.	**Unit 3: Two Dimensional Geometric Transformations and Viewing (8 hrs)** 3.1 Two Dimensional Transformations 3.1.1 Translation, Rotation, Scaling, Shearing, Reflection 3.2 Matrix Representations of Transformations 3.3 Homogeneous Coordinate System 3.4 Composite Transformations 3.5 Windowing Concepts, Two Dimensional Viewing Pipeline 3.6 Window to Viewport Transformation 3.7 Line Clipping Algorithm: Cohen-Sutherland 3.8 Polygon Clipping: Sutherland-Hodgeman
Learn about the extensions of two-dimensional methods for geometric transformation in three dimensions. Helps them apply the knowledge of three dimensional rendering and transformations for representing objects more accurately	**Unit 4: Graphics in Three Dimensions (8 hrs)** 4.1 Three Dimensional Coordinate Systems 4.2 Three Dimensional Transformations 4.2.1 Translation, Rotation, Scaling, Reflection 4.3 Three Dimensional Representations 4.3.1 Polygon Surfaces 4.3.2 Cubic Spline and Beizer Curve 4.3.3 Non-Planer Surface: Bezier Surface 4.3.4 Fractal Geometry Method 4.4 Three Dimensional Viewing Transformation and Pipeline 4.5 Projections 4.5.1 Parallel Projection 4.5.2 Perspective Projection 4.6 Clipping in Three Dimensions
Get acquainted with the techniques used for bringing visual realism in computer generated graphical output primitives. Familiarize with visible surface detection techniques for creating realistic displays and in computing	**Unit 5: Visual Realism (7 hrs)** 5.1 Hidden Surfaces and Hidden Surface Removal Approaches 5.1.1 Back-Face Detection 5.1.2 Depth Buffer Method 5.1.3 A- buffer method 5.1.4 Scan Line Method 5.1.5 Depth Sorting Method

light intensities of surfaces using lighting or shading models for generating realistic views	5.2 Illumination and Shading Methods 5.2.1 Illumination Theory and Models 5.2.1.1 Ambient Light 5.2.1.2 Diffuse Reflection 5.2.1.3 Specular Reflection (Phong Model) 5.2.2 Polygon Surface Shading Methods 5.2.2.1 Constant Shading (Flat Shading) 5.2.2.2 Gouraud Shading 5.2.2.3 Phong Shading 5.2.2.4 Fast Phong Shading 5.3 Color Models 5.3.1 RGB 5.3.2 CMYK
Gain a good understanding of the use of graphical standards necessary for achieving software portability. Become familiar with the basic structure or format of a graphical file and know the ways of visually representing data sets. Learn about writing graphical programs that can be ported to multiple hardware platforms without extensive rewriting of the programs through the use of graphical standards.	**Unit 6: Graphical Standards (4 Hrs)** 6.1 Need for Machine Independent Graphical Languages 6.2 Graphical Standards: PHIGS, GKS 6.3 Graphics Software Standards and Language Binding 6.4 Overview of Graphics File Formats 6.5 Visualization of Data Sets
Learn about the library functions provided in OpenGL for graphical rendering and performing geometric transformations. Enable students to write programs using functions in OpenGL library for drawing graphical output primitives and performing basic transformations	**Unit 7: Introduction to OpenGL (4 Hrs)** 7.1 Overview and Basic Architecture of OpenGL 7.2 GL Related Libraries 7.2.1 Drawing Basic Output Primitives and Polygons 7.2.2 Call Back Functions and Input Handling 7.2.3 Basic Transformations 7.2.4 Projections and Lighting

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Solving problems related to Raster Graphics system, e.g. computation of frame buffer size, color manipulation techniques, aspect ratio, refresh rate, resolution etc
2	Identifying the points for digitizing line, circle and ellipse using the algorithms studied

3	Solving problems related to Two Dimensional Transformations and Matrix Compositions like fixed point scaling, pivot point rotation, reflection about arbitrary line etc
4	Mapping of object descriptions from window to viewport and clipping of line segments against a clip window
5	Solving problems related to Three Dimensional Transformations
6	Computing points necessary for constructing a Bezier Curve with the given set of control points and number of line segments
7	Solving problems related to Parallel and Perspective Projection
9	Determining surface normal of polygons and determining visibility using Back face detection approach
10	Determining averaged intensity at a point of a polygon using Gouraud Shading

6. Practical Work

Students are required to implement the 2D and 3D graphics algorithms studied in the course using C/C++ and OpenGL.

Students are required to demonstrate the assimilated knowledge in Computer Graphics through project work (Games, Graphical Simulations) submitted at the end of the semester. Students need to form a team of 4-5 members and they are allowed to explore new programming languages or platforms (Unity, Unreal Engine or any related tools) for accomplishing the project work.

S.N.	Practical works
1	To familiarize students with the tools to be used for implementing algorithms studied in the course
2	To draw a straight line using Digital Differential Analyzer and Bresenham's Line Drawing Algorithm
3	To digitize a circle using Circle Drawing Algorithm
4	To digitize an ellipse using Ellipse Drawing Algorithm
5	To draw Two Dimensional objects on the screen and perform two dimensional transformations to them
6	To fill objects using Boundary Fill and Flood Fill approaches
7	To draw the projected view of a Three Dimensional object using perspective and parallel projection and perform three dimensional transformations to it
9	To draw a smooth curve using Bezier curve with designated control points
10	To draw graphical output primitives using OpenGL, perform various transformations and implement algorithms covered in the course

7. Evaluation system and students' responsibilities

Internal Evaluation

In addition to the formal end-semester exam(s), the internal (formative) evaluation of a student may consist of quizzes, assignments, lab reports, projects, class participation and

presentation etc. The tabular presentation of the internal evaluation is as follows. The components may differ according to the nature of the subjects.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	20%			
Lab Report/Project Report	30%			
Practical Exam/Project Work	30%			
Viva	20%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student requirements:
Each student must secure at least 45% marks in internal evaluation with 80% attendance in the class in order to appear in the semester-end examination. Failing to get such a score will be equated with NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End- Semester examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. Failure of a student to attend a formal exam, quiz, test, etc. won't qualify him/her for re-exam. ***Students are required to complete all the requirements defined for the completion of the course***

8. Prescribed Books and References

Prescribed Text Book

Donald Hearn and M. Pauline Baker: Computer Graphics C Version, Prentice-Hall.

Reference Books

Donald Hearn and M. Pauline Baker: Computer Graphics with OpenGL, Prentice-Hall.

James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics: Principles and Practice in C, Addison-Wesley

Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, "OpenGL Programming Guide", The Official Guide to Learning OpenGL, OpenGL Architecture Review Board, LPE Pearson Edition Asia