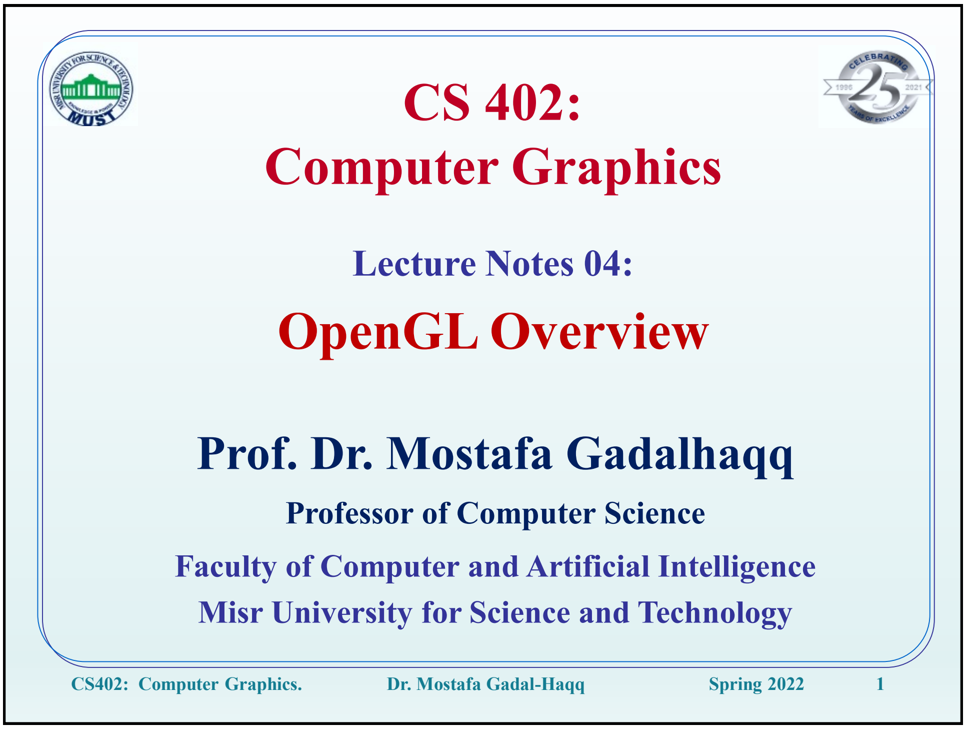


CS 402: Computer Graphics

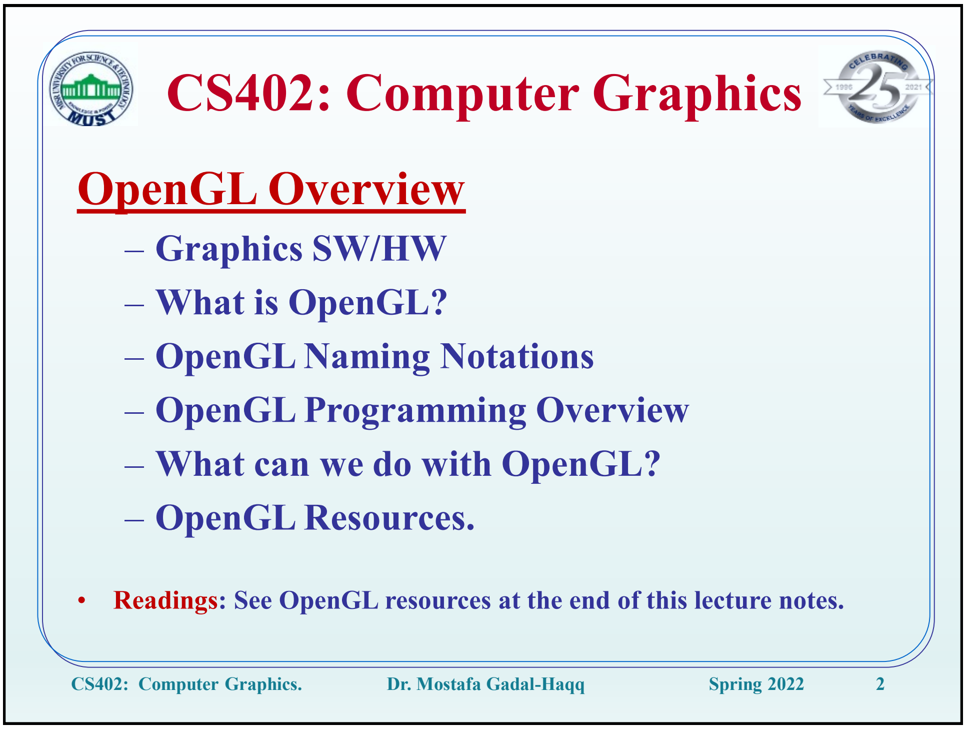
Lecture Notes 04:

OpenGL Overview

Prof. Dr. Mostafa Gadalhaqq

Professor of Computer Science

Faculty of Computer and Artificial Intelligence
Misr University for Science and Technology



CS402: Computer Graphics

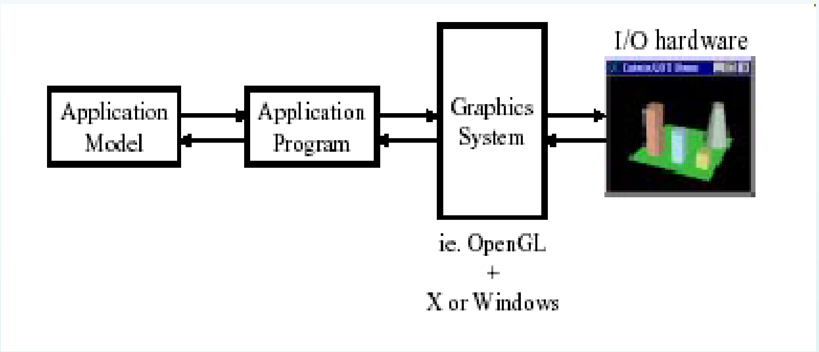
OpenGL Overview

- Graphics SW/HW
- What is OpenGL?
- OpenGL Naming Notations
- OpenGL Programming Overview
- What can we do with OpenGL?
- OpenGL Resources.

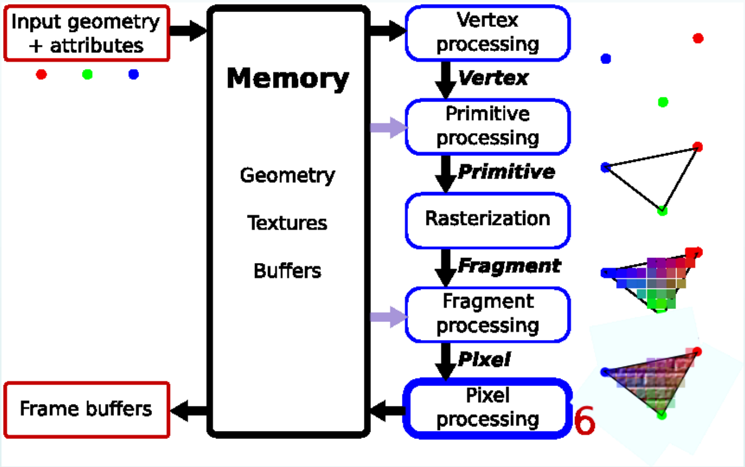
- **Readings**: See OpenGL resources at the end of this lecture notes.

Graphics SW/HW

Overview of a SW Graphics application

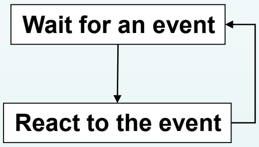


OpenGL Pipeline



Interactive Programming

- Interactive programs are event driven



- What are events?
  - Mouse Click, double click, hover, etc.
  - Button pressed
  - Time elapsed

What is OpenGL?

- OpenGL: Open Graphics Library
  - the premier environment for developing portable, interactive 2D and 3D graphics applications.
  - Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms.
  - fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions.

What is OpenGL?

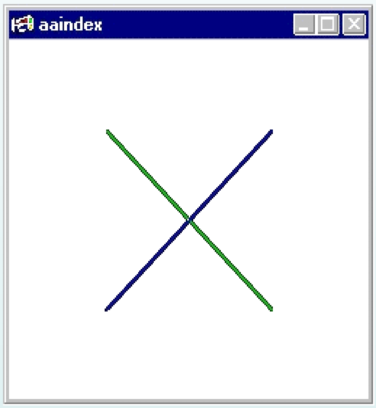
- A software interface to graphics hardware.
  - Contains routines that can be used to specify objects and operation needed to produce 3D application.
- Platform independent
  - No support for window tasks (see GLUT).
  - Use geometric primitives to construct 3D models (objects).

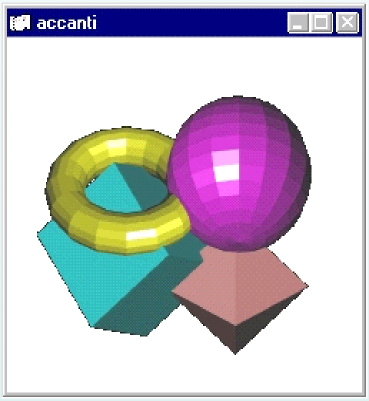
What is GLUT?

- GLUT: OpenGL Utility Toolkit
  - A Graphics library that handle window tasks
    - Create windows
    - detect user input (Keyboard strokes, mouse clicks)
  - Also contains some complex 3D shapes.
    - A sphere, a torus, and a teapot

OpenGL Examples

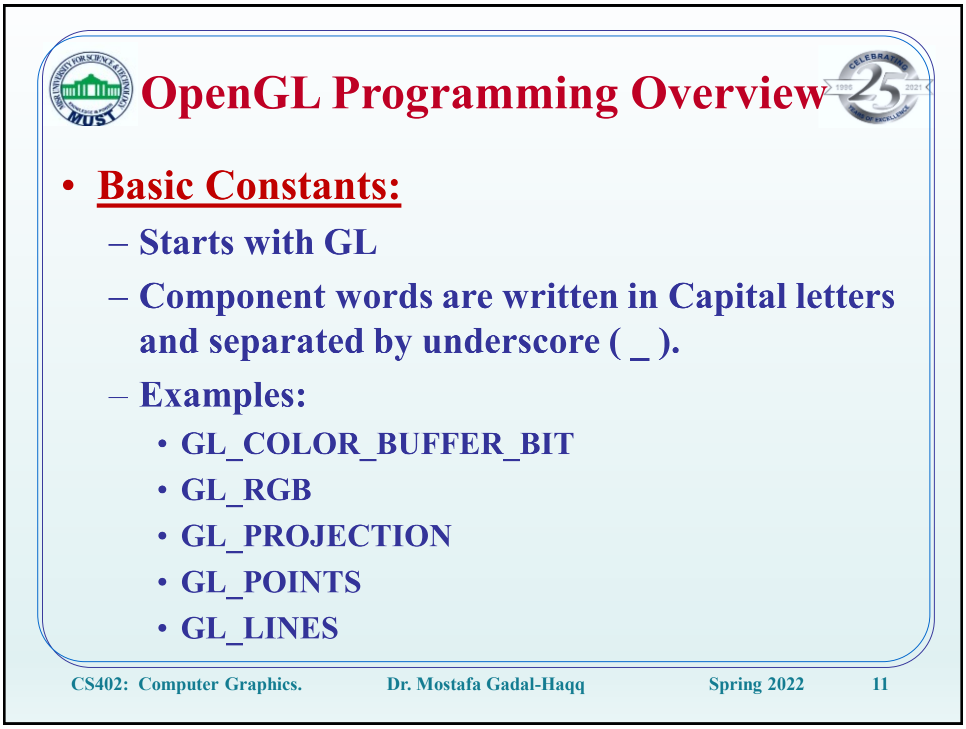
- Some examples of OpenGL graphics





OpenGL Programming Overview

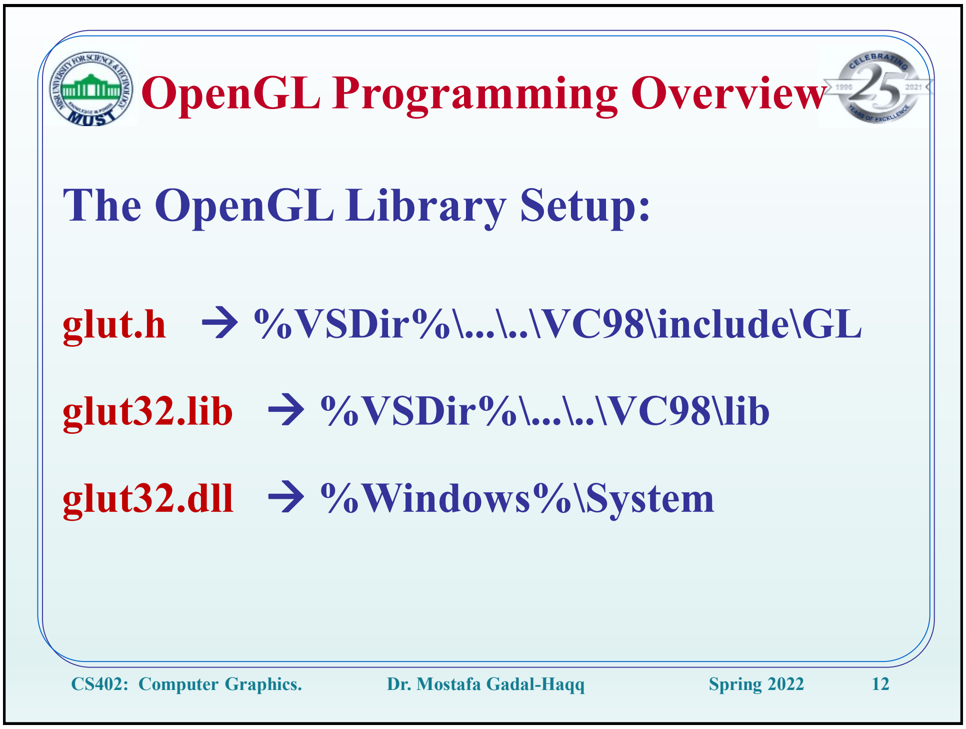
- OpenGl is a library of functions (Not OOP):
  - Function names:
    - Begins with gl .
    - Each component word start with capital letter.
    - Some functions require constant arguments.
  - Examples:
    - glClear()
    - glClearColor()



OpenGL Programming Overview

- **Basic Constants:**
  - Starts with GL
  - Component words are written in Capital letters and separated by underscore (_).
  - Examples:
    - GL_COLOR_BUFFER_BIT
    - GL_RGB
    - GL_PROJECTION
    - GL_POINTS
    - GL_LINES

CS402: Computer Graphics. Dr. Mostafa Gadal-Haqq Spring 2022 11



OpenGL Programming Overview

The OpenGL Library Setup:

glut.h → %VSDir%\...\..\VC98\include\GL

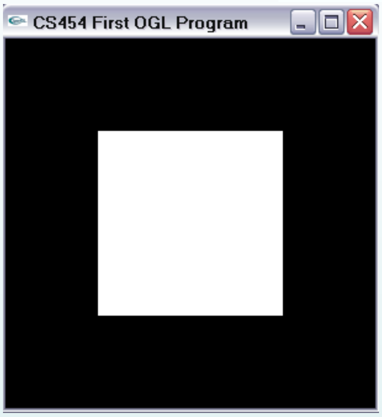
glut32.lib → %VSDir%\...\..\VC98\lib

glut32.dll → %Windows%\System

CS402: Computer Graphics. Dr. Mostafa Gadal-Haqq Spring 2022 12

OpenGL Programming Overview

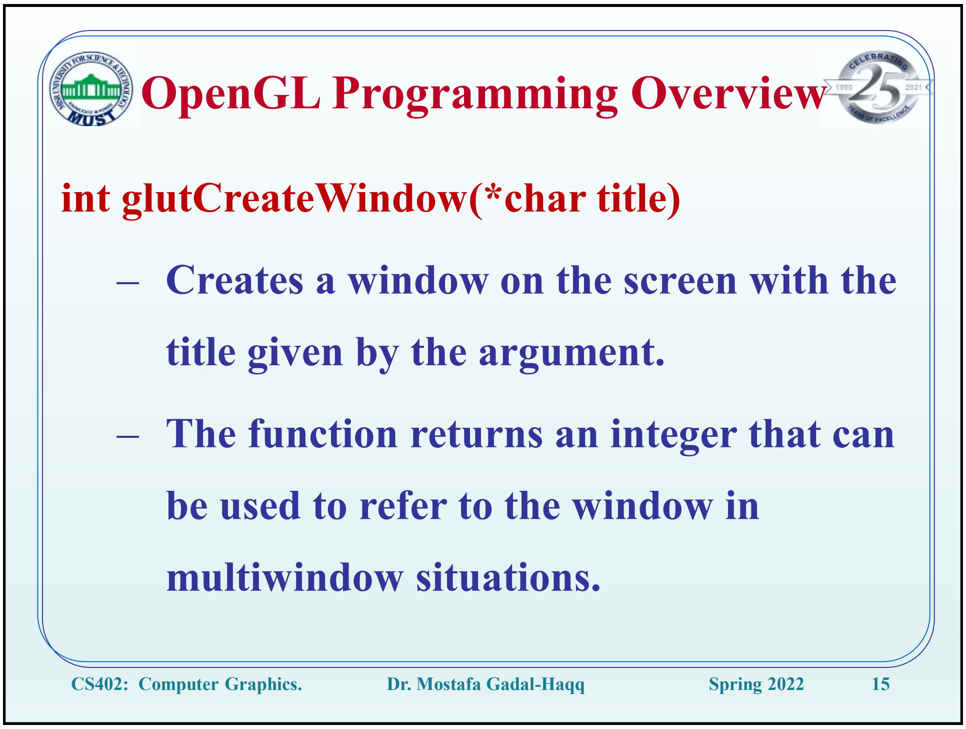
```
#include <GL/glut.h>
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
        glVertex2f(-0.5, -0.5);
        glVertex2f(-0.5, 0.5);
        glVertex2f(0.5, 0.5);
        glVertex2f(0.5, -0.5);
    glEnd();
    glFlush();
}
int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutCreateWindow("CS454 First OGL Program");
    glutDisplayFunc(display);
    GlutMainLoop();
}
```



OpenGL Programming Overview

void glutInit(int argc, char **argv)

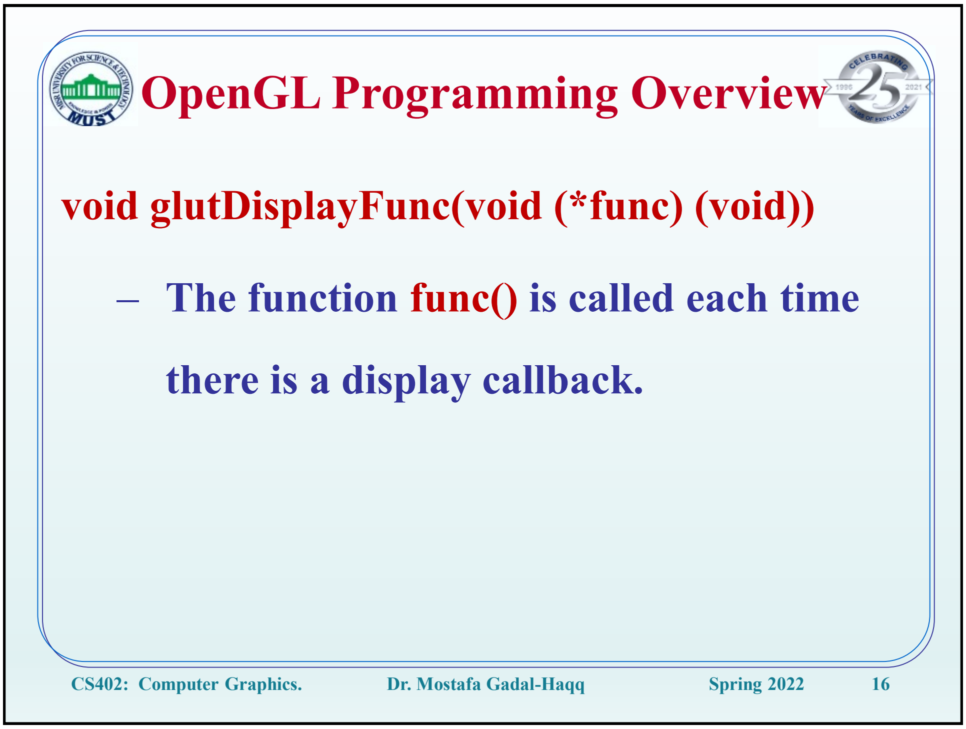
- Initializes GLUT and should be called before any OpenGL functions: glutInit() takes the arguments from main() and can use them in an implementation dependent manner.



OpenGL Programming Overview

int glutCreateWindow(*char title)

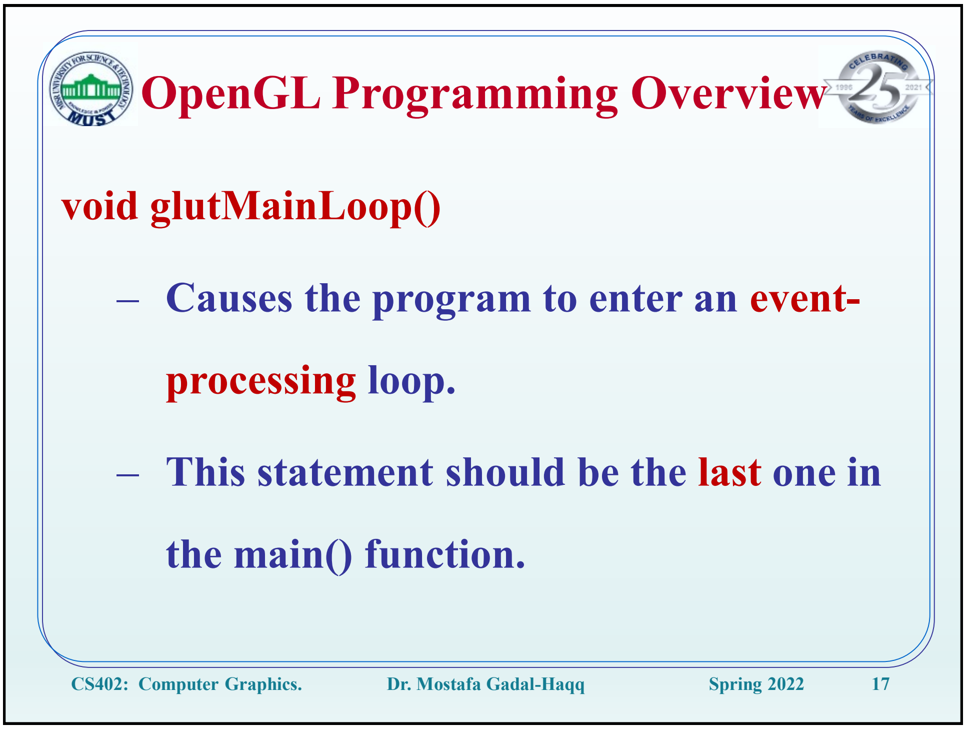
- Creates a window on the screen with the title given by the argument.
- The function returns an integer that can be used to refer to the window in multiwindow situations.



OpenGL Programming Overview

void glutDisplayFunc(void (*func) (void))

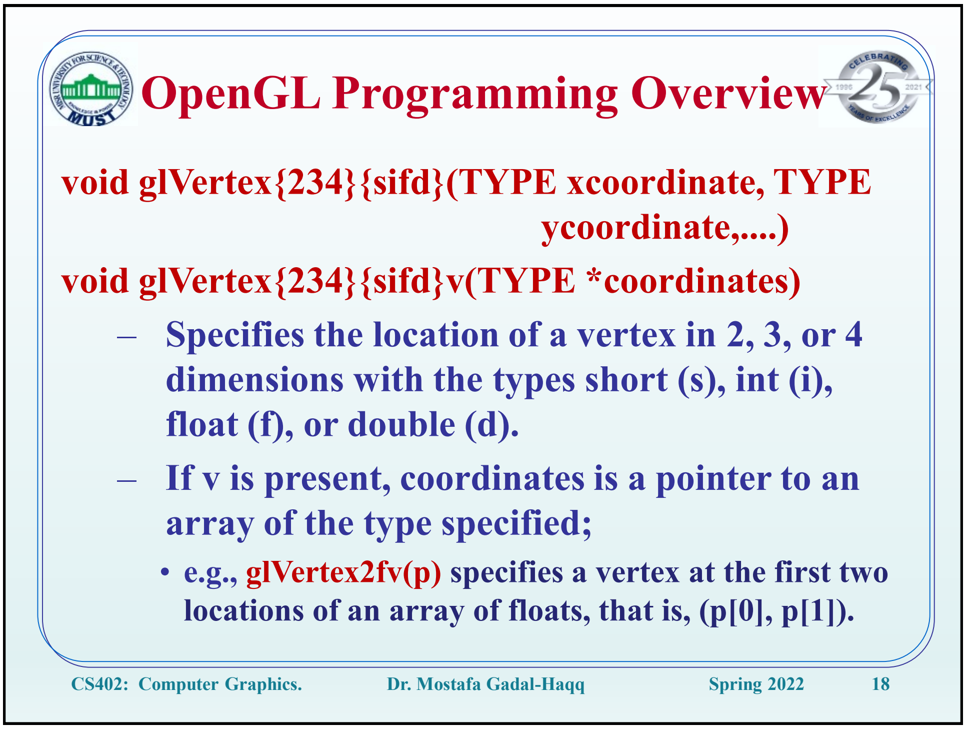
- The function **func()** is called each time there is a display callback.



OpenGL Programming Overview

void glutMainLoop()

- Causes the program to enter an event-processing loop.
- This statement should be the last one in the main() function.

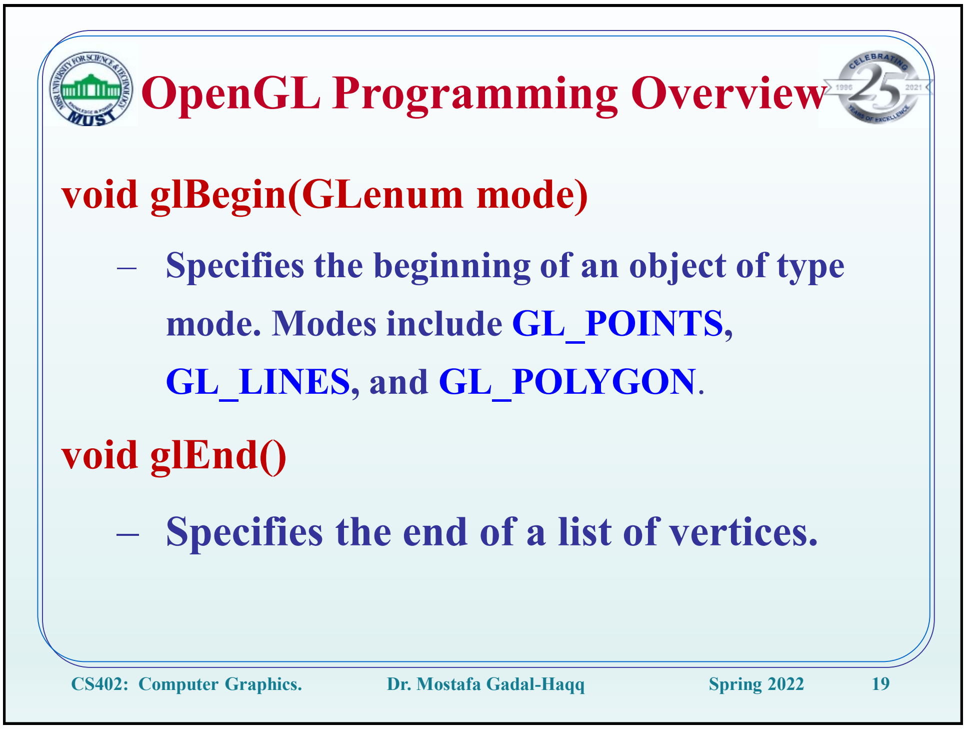


OpenGL Programming Overview

void glVertex{234}{sifd}(TYPE xcoordinate, TYPE ycoordinate,....)

void glVertex{234}{sifd}v(TYPE *coordinates)

- Specifies the location of a vertex in 2, 3, or 4 dimensions with the types short (s), int (i), float (f), or double (d).
- If v is present, coordinates is a pointer to an array of the type specified;
  - e.g., glVertex2fv(p) specifies a vertex at the first two locations of an array of floats, that is, (p[0], p[1]).



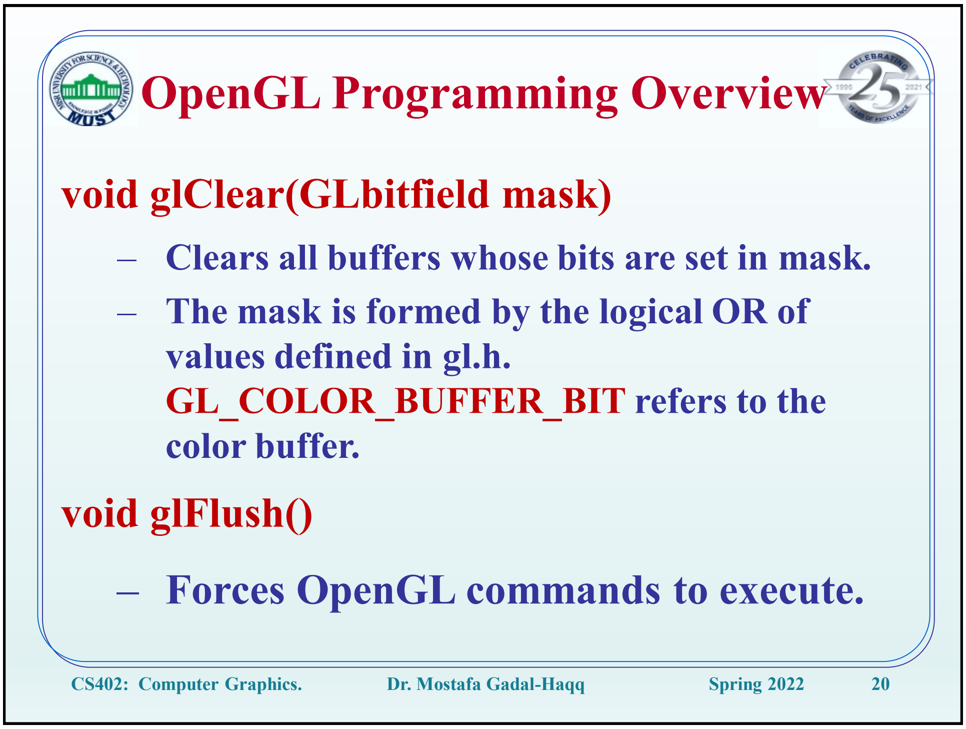
OpenGL Programming Overview

void glBegin(GLenum mode)

- Specifies the beginning of an object of type mode. Modes include GL_POINTS, GL_LINES, and GL_POLYGON.

void glEnd()

- Specifies the end of a list of vertices.



OpenGL Programming Overview

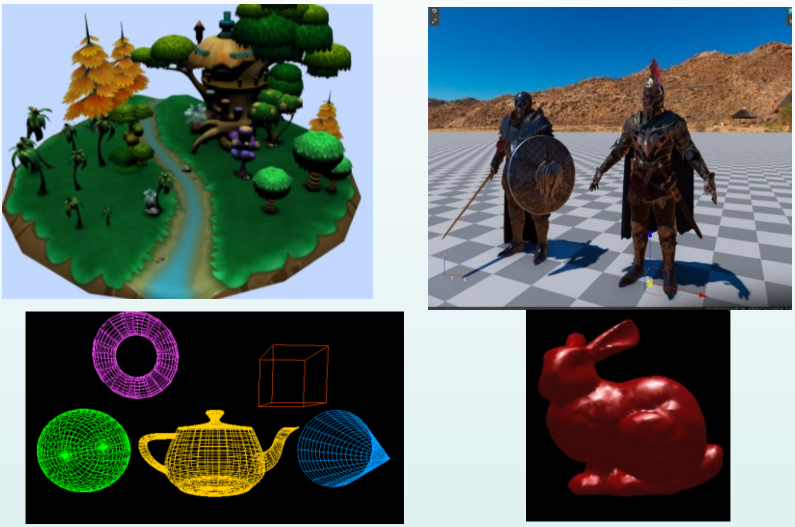
void glClear(GLbitfield mask)

- Clears all buffers whose bits are set in mask.
- The mask is formed by the logical OR of values defined in gl.h. GL_COLOR_BUFFER_BIT refers to the color buffer.

void glFlush()

- Forces OpenGL commands to execute.

What can we do with OpenGL?



OpenGL Resources

- **OpenGL Tutorial:**
  - https://www.opengl-tutorial.org/beginners-tutorials/
- **OpenGL in One Lecture:**
  - https://www.youtube.com/watch?v=45MIykWJ-C4
- **Every thing about OpenGL in a 31-Lectures:**
  - https://www.youtube.com/watch?v=W3gAzLwfIP0&list=PLlrATfBNZ98foTJPJ_Ev03o2oq3-GGOS2&index=1
- **OpenGL Projects:**
  - https://www.codeproject.com/Articles/1186329/OpenGL-MFC-Projects-in-One

Practice OpenGL and Have Fun!