Advanced Functions

void iKeyboard (unsigned char key)
Description: called whenever the user hits a key in keyboard. **Parameters**: key- holds the ASCII value of the key pressed.
void iSpecialKeyboard (unsigned char key)
Description: called whenever user hits special keys like- function keys, home, end, pg up, pg down, arrows etc. you have touse appropriate constants to detect them. A list is: GLUT_KEY_F1, GLUT_KEY_F2, GLUT_KEY_F3, GLUT_KEY_F4, GLUT_KEY_F5, GLUT_KEY_F6, GLUT_KEY_F7, GLUT_KEY_F8, GLUT_KEY_F9, GLUT_KEY_F10, GLUT_KEY_F11, GLUT_KEY_F12, GLUT_KEY_LEFT, GLUT_KEY_UP, GLUT_KEY_RIGHT, GLUT_KEY_DOWN, GLUT_KEY_PAGE UP, GLUT_KEY_PAGE DOWN, GLUT_KEY_HOME, GLUT_KEY_END, GLUT_KEY_INSERT **Parameters**: key- holds the ASCII value of the key pressed.
void iMouse (int button, int state, int mx, int my)
Description: called when the user presses/releases the mouse.(mx, my) is the position where the mouse pointer is. **Parameters**: button: GLUT_LEFT_BUTTON, GLUT_RIGHT_BUTTON state: GLUT_DOWN, GLUT_UP mx,my - coordinate of mouse pointer
void iMouseMove (int mx, int my)
Description: called when the user presses and drags the mouse. (mx, my) is the position where the mouse pointer is. Pressing the mouse buttons won't have any effect here. **Parameters**: mx,my - coordinate of mouse pointer
void iPassiveMouse (int mx, int my)
Description: is called when the user moves the mouse. (mx, my) is the position where the mouse pointer is. **Parameters**: mx,my - coordinate of mouse pointer

int iLoadImage (char filename[])
Description: returns an integer id for the image specified in filename. The id will be used later for rendering (like file pointer). This function needs to be invoked in the main function. It supports png or jpeg formats unlike iShowBMP(); **Parameters**: filename[]: name of the file. The directory must be mentioned unless the image is in the project directory. **Example**: If the image is in the folder called "image" which is inside theproject folder, we use: iLoadImage("image\\samplepic.png);
void iShowImage (int x, int y, int width, int height, unsigned int id)
Description: render the image of specific id obtained from previous function. Image will be rendered at (x,y) with width and height (like iRectangle()). So you can scale the image by tweaking width and height. **Parameters**: x, y: coordinate of the point where the image will be placed width, height: size of the image to scale id: the integer value obtained from iLoadImage() call.
void iRotate (double x, double y, double degree)
Description: Rotates the coordinate system **Parameters**: (x, y) - The pivot point for rotation degree - degree of rotation
void iUnRotate()
Description: reset the rotation made by the function iRotate(); **NOTE**: After calling iRotate(), every subsequent rendering will happen in rotated fashion. To stop rotation of subsequent rendering, call iUnRotate().

```
void iShowImageSmart(int x, int y, unsigned int texture, double ratio=1)
```

Description: Renders the image of specific id obtained from iLoadImage() by automatically calculating the height & width of the image. The fourth parameter can be used to change the size of the image while keeping the aspect ratio fixed.

Parameters:
x,y: coordinate of the point where the image will be placed
id: the integer value obtained from iLoadImage() call.
ratio: optional parameter with default value of 1. Passing 2 will make the image half in size.