ETGG3802	Lab01: **ssurge Component system**	Points: **100**
Assigned: **1/12/2016**	Due: *no sooner than* **1/20/2016 @ about 3:00pm**	

Overview:

- Our GameObject class is quickly becoming cluttered – we need to break up the logic.
- I'm modelling this after Unity: GameObjects are basically a scene node. You "snap-in" Components to add functionality (e.g. MeshRenderer, 3dSoundListener, First-person-controller, etc.)
- In this lab, we'll make a Component class and as a test, move mesh functionality to a new MeshComponent class.

Future Improvements:

- Add more functionality to MeshComponent (e.g. bone-based animation support)
- Add many more types of Components
- Incorporate scripting
- A Manager for each type of component (a "pool" of related components could increase cache coherency)

Tasks:

1. Download from the SVN repo (in this lab, I want you to use my solution – this will be the starting point for our ssurge group work)
    a. If anyone wants to port part of their lab over to mine, I'm willing…(get permission first)
2. (**20 points**) **Component class**: Include the following pure-virtual methods
    a. std::string getTypeName()
    b. ComponentType getType()
        i. Make this enumeration class in Component.h
    c. void update(float dt) // This could be "normal" virtual (make an empty body)
    d. A GameObject * (the game object that "owns" this component).
    ~~e. These two methods can probably be "normal" virtual~~
        ~~i. initialize() // Called by constructor (and others?)~~
        ~~ii. shutdown() // Called by destructor (and others?)~~
3. (**20 points**) **MeshComponent**
    a. Derive from Component
    b. Move the Ogre::Entity * from GameObject here (if there is one)
    c. Define all pure-virtual methods
    d. Make the process of attaching components as error-proof as possible (generally the fewer # of steps the user needs to do, the better).
4. (**30 points**)**GameObject** changes
    a. Make a container [listen in class – we'll decide on the form] of Component*'s
    b. Add functionality to create and set up a mesh component
5. (**15 points**)**GOM** changes
    a. Modify what we do when we encounter a 'mesh' part of the xml file.
6. (**15 point**s) Test code that fully tests all new functionality.

*Submit a .zip of **just** your source and include directories (make sure everything is there!)*