CSC-133 (Fall 2018)
Attendance Quiz 1 – OOP Concepts and UML Class Diagram

Student Name: Warren Quattrocchi

Question 1: Define encapsulation, abstraction. Explain how these two concepts relate to object-oriented programming (OOP) **(10 points).**

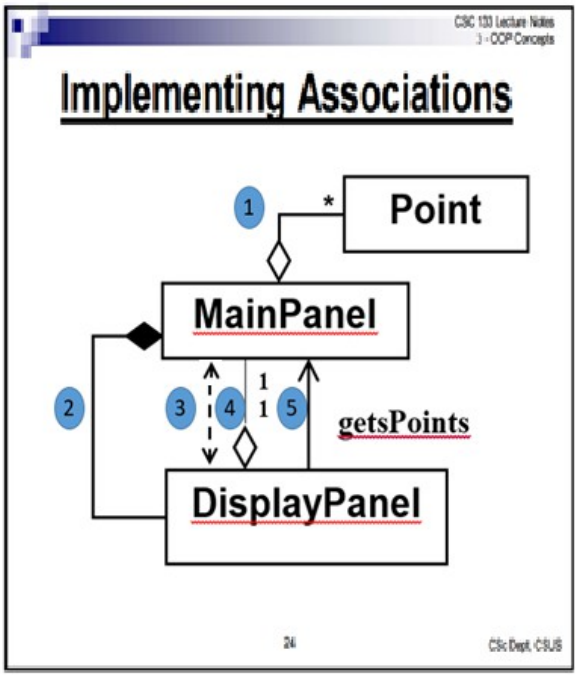
Encapsulation is a way to keep relevant data and methods together thorugh "bundling" and "information hiding"

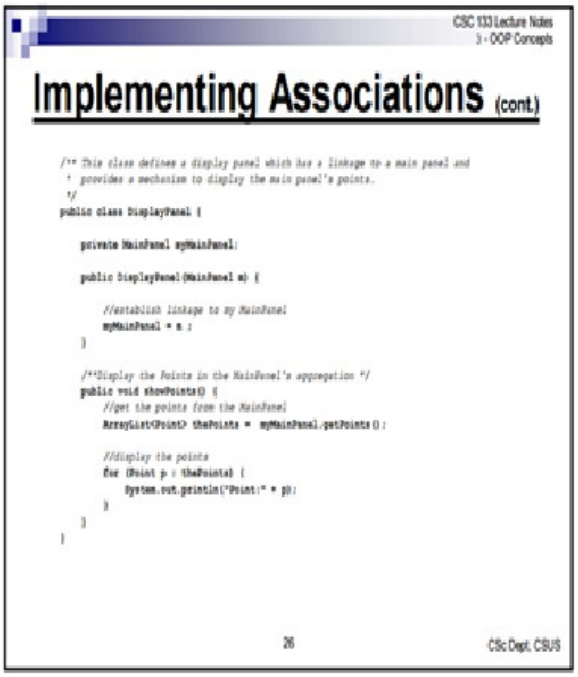
- Encapsulation is important in object-oriented programming as it helps keep related objects together and from unrelated objects

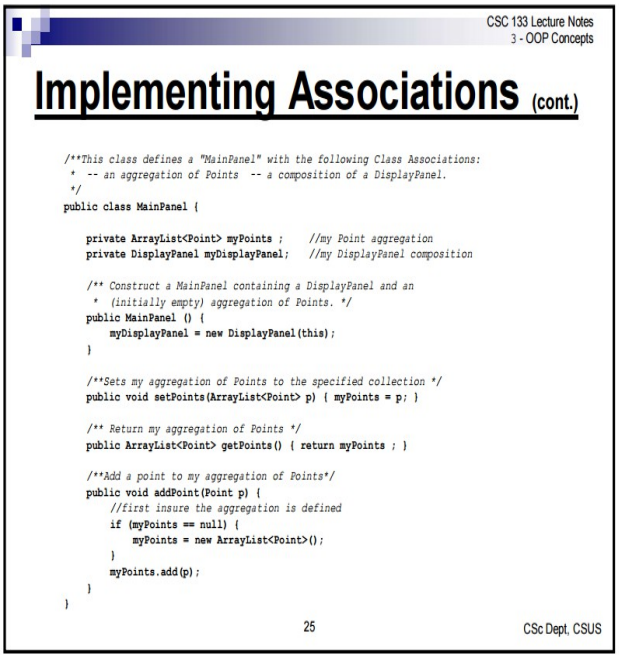
Abstraction is the process of removing characteristics from something in order to reduce it to a set of essential characteristics, and identify these objects by essential characteristics.

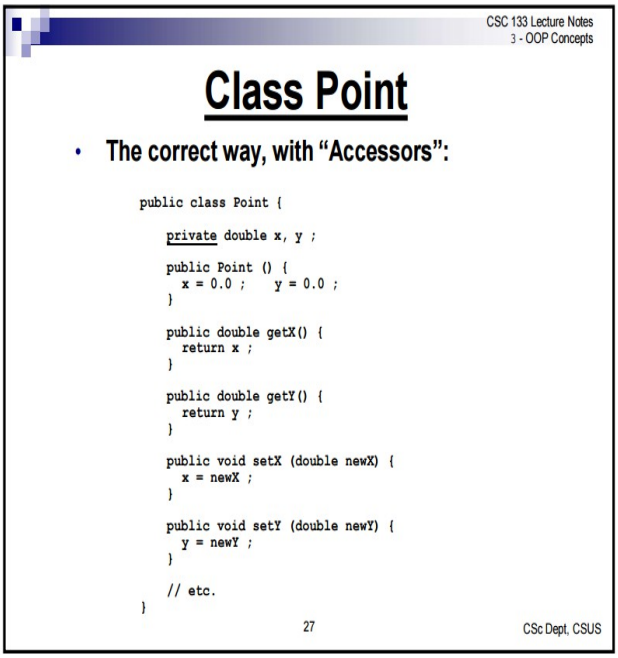
-Abstraction is important in object-oriented programming as it can increase the amount of re-usable code and greatly simplify systems.

Question 2: UML is not just about pretty pictures. If used correctly, UML precisely conveys how code should be implemented from diagrams. If precisely interpreted, the implemented code will correctly reflect the intent of the designer. Please review the following slides. In the class diagram in slide 24, given the labels 1, 2, 3, 4, and 5 please fill out the required information in the table beneath. **BE SURE TO JUSTIFY YOUR ANSWER. (10 points).**









Label Number	Name the association	Justifications (Please use the "Recap 1" Table for reference)	Specify Java Class(es) Name and Line of codes
1	Aggregation	A MainPanel **has** Points Points can exist without MainPanel	**In class MainPanel:** private ArrayList<Point> myPoints;
2	Composition	A MainPanel is **made up of** a DisplayPanel The DisplayPanel cannot exist without MainPanel	**In class MainPanel:** private DisplayPanel myDisplayPanel;

3	Dependency	MainPanel contains parameters of type DisplayPanel
		
		DisplayPanel contains parameters of type MainPanel
			In class MainPanel:
			public MainPanel(){
			myDisplayPanel = new DisplayPanel(this);
			}
			In class DisplayPanel:
			public DisplayPanel(MainPanel m) {
			myMainPanel = m;
			}

4	Aggregation	A DisplayPanel **has a** MainPanel	**In class DisplayPanel:** private MainPanel myMainPanel;
5	Association	DisplayPanel sends a message to MainPanel	**In class DisplayPanel:** ArrayList<Point> thePoints = myMainPanel.getPoints();