-Listing potential classes, actions/methods/relationships (verbs)

-Drawing UML class diagrams

-Deriving Use Cases

-Draw Sequence Diagram

-Identify Design Patterns based on picture (draw design the design patterns)

-Git  
 -How to track staged delivery process where clients might be using older (stabler) versions

-Human Error and Usability   
 -Relationship between **iterative model** and **waterfall model**. What is their primary difference?

-Name of the law that describes the **speed of choosing from a list of choices**

-Why is the **difference in time of choosing 2 and 8** choices greater than the difference between 80 and 100

-What is **Saccadic Masking** and how does it affect software

-User Interfaces  
 -What is one UI method that aids usability but also reduces human error?

-Why must we be careful about **colours** we use in UI? Colour blindness

-Give 2 examples (or instances) of interface metaphors.

-Identify Design Patterns appropriate for (and explain why):  
 -Want to implement macros learned from the user. These macros can be stored and replayed later.

-Event-based system where users can add plugins at run-time. These plugins can agree to handle some events but might only do so conditionally

-Making a program that procedurally details a universe lazily. Can go down from galaxies to solar systems to planets to countries to people to cells to atoms etc

-Refactoring  
 -Find at least 3 bad smells, and at least 1 refactoring that could be applied to this code snippet. Then DRAW the UML class diagram of the code after refactoring

-Testing  
 -Write a class for a **mock object** that will allow for testing of line x of xClass

Clear:

2012F