**Plan of Attack**

**Step 1: I/O**

* Implement the observer pattern on our Floor, TextDisplay, and Cells (EmptySpace, Wall, Passage, Door, Tile).
* Concurrently, implement a text reading mechanism that allows us to read in floor plans from a file
  + This will be done through command line arguments as well as fstream input
  + Each character in the file will be converted into a Cell of the appropriate type
  + These Cells will then be added to our Floor class
  + Floor will then notify TextDisplay what it should have at each of its positions (by calling the notifyTextDisplay() function as part of each Cell in the Floor)
* Once we have a working input method, we will print out the TextDisplay using a operation overload on << with Floor and TextDisplay
* This will finalize our I/O for floor plans

**Step 2: PC Movement**

* Implement basics of Thing, Character, PC, and Human classes
  + Add HP, Attack, Defense, Race, and Money to the screen
* Create Chambers and link them to Floor
* Generate PC in a chamber
* Create movement functions for PC
* Notify Floor for every time a PC moves
* Reprint the TextDisplay for every move

**Step 3: Items on Board**

* Create Item abstract class
* Create Gold abstract class
* Create NormalGold Class
* Create Potion abstract class
* Create RestoreHealth Potion
* Generate 10 RestoreHealth potions and 10 NormalGold gold in the Chambers at the beginning of a Floor
* Make sure 2 Items don’t generate on the same space

**Step 4: Enemies on Board**

* Create Enemy abstract class
* Create Goblin class
* Generate 20 Goblins throughout the 5 chambers
* Make sure 2 Items/Enemies don’t generate on the same space

**Step 5: Item Interactions**

* Implement Potion’s drinkPotion() and PC’s discoveredRH() functions
* Implement stepping on Gold which triggers the getAmount() function
* Ensure items disappear from Floor every time an Item is interacted with
* Make text actions for seeing and drinking Potions

**Step 6: Enemy Movement**

* Implement a step function that allows Enemies to move around the board every time the player interacts with the world (stepEnemy() function)

**Step 7: Combat**

* Allow PC to attack people in any direction (1 block radius)
* Add option for Enemies to attack within the stepEnemy() function
* When Enemies die, increase the PC Money amount
* Create combat text actions

**Step 8: Stairs**

* Generate Stairs, ensuring that PCs and Stairs are not generated in the same chamber