1.
Always have a plan: Generally speaking it's rare to not have any form of plan, however during the "make a game" assignment for 1005, while I did have an overarching plan of "make a game" the specifics somewhat eluded me, resulting in a project that just met the requirements, was submitted at the absolute last second, and frankly I wasn't happy with it.

Look for analogies: Not always applicable. So far, nothing terrible has happened by not applying this problem solving technique.

2.
step one: research - what are the specifics of dominos, how can we best represent it as a struct
step two: simplify - construct a simple domino struct with preset values (ex. all sixes)
step three: simplify other components - create a basic implementation of the domino display code
step four: complicate slightly - construct an array of the simple domino structs
step five: slightly complicate other components - modify the display code to show the dominos from the array
step six: enhance - allow the domino structs to have the needed non-fixed values, and double check the display code to ensure it still works as expected
step seven: complicate more - write domino sorting code, prove it works by outputting the fancy new sorted dominos
step eight: test and refine - see how the whole program works, fixing any edge cases or bugs
step nine: ???
step ten: profit

divide/reduce the problem: take portions of the problem, reduce them to their base components, and add in features and functionality as needed.
experiment: build a variety of prototypes of functionality and refine and iterate.