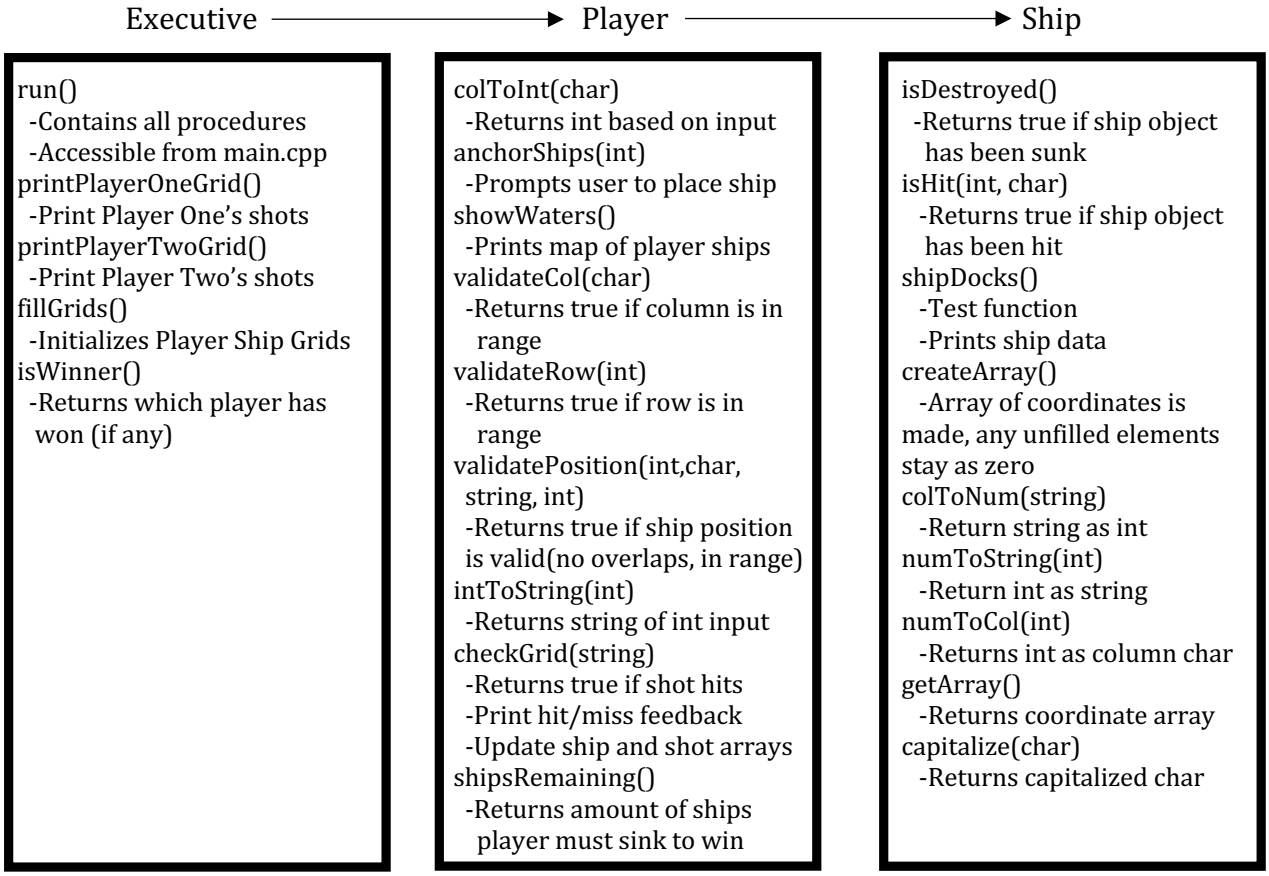
EECS 448 – Project 2 – Battleship – Team 5

The team who originally designed the Battleship game our team inherited with Object-Oriented Design. There is a top-down functional decomposition approach, in which the Executive class controls the game's procedures, the Player class validates user inputs for ship placement, and the Ship class contains relevant data about a single ship. The Executive object has access to relevant public-facing Player methods and the Player object has access to relevant public-facing Ship methods.



Our team recognized some design flaws in the original implementation of the program. Many of these issues stem from methods being irrelevant to the class, avoidable, or repetitive, especially in the Executive object in which too much responsibility is put on the run method and the printPlayerOne/TwoGrid methods are repetitive and irrelevant.

In our team's iteration, we corrected these errors by dividing the Executive object's run method's responsibility into several digestible methods, including a takeTurn method which called several of the current player's methods – printFiringBoard (this replaces print the aforementioned PlayerOne/TwoGrid method), printShipBoard (showWaters method with more clear name), and takeShot (a new Player method, largely taken from code originally found in the Executive run method).