Akin Omisakin report:

To run code use -jar assignment1-1.0-SNAPSHOT.jar 4 4 1 this means a 4 X 4 grid and 1 ship are created each are 3 spaces long

Task 1_1: after I did set name in constructor then used to the Random () to randomly pick between 0- 1 0 for vertical 1 for horizontal

Task 1_2: I implemented the inherited methods from Abstract battleship straight forward not much time spent.

Task 1_3: check Attack was the took the longest because I had to search ship Coordinates for match values as the method parameters

Task 2_1: To populate the game Grid with grid dots required double for loops

Task 2_2: Converted the counting number (Starting from 1) by adding it to "Ship" String then storing each one into ships

Task 2_3: first step was to get the random values I decided to split the random numbers into 4 to make it easy

second step respect the horizontal and vertical orientation by working out add each value to the temporary array

third step equal game Grid random coordinates to '*' then setShipCoordinates = tempArray

Task 2_4: extend GameGrid to PlayerGameGrid and OpponentGameGrid. Need to use print instead of print and println to form grid structure. Hide ships on OpponentGameGrid by replacing them with '.'. Everything else showed up

Task 3_1: inherited the GameControls to Game then implemented the methods. return type of getOpponentssGrid and getPlayersGrid had to be the same a AbstractGameGrid so I created 2 object fields and returned them

Task 3_2: used System.exit(1) to shut code when condition was met, or it would loop if another string were entered

Rask 3_3: used two local variables win and lose to determine what the output would be if I won the game or lost if returned true then the appropriate message would print.

used the || to return both Boolean variables, do not work with && since they can be different values.

Task 3_4: definitely the hardest one, not much success when trying to do both grid under one loop so slit them up. First thing I did was set values in local variable so I could control it better.

I had a trouble getting a good accuracy using only one shipCoordinates of a single '*' so I added 2 more. then did the same with the miss because there was an error were only miss ('%') would appear on the grid.

exitGame () had to be at the start of playround so minimize errors in the game.

Task 4_1: used 1 instance of game that require parameters when run.

Task 4_2: code runs continuously as long checkVictory = false.

Task 4_3: if user inputs exit code will shut down. also, if checkVictory = true which it does when ALL ships are destroyed, they print "you have won" or "lost"

Task 4_4: input is passed into PlayedRound(input) - > localString

Task 4_5: Incorrect input using input mismatch exception