Testing move(row,column)	Input Value	Expected Value
Move played at row 2 column3	Move(2,2)	"Pass"
Values of row and column in the needed grid	Move(8,9)	"IndexOutOfBound"
Negative row values	Move(-1,2)	"IllegalArguementException"
Negative column values	Move(1,-2)	"IllegalarguementException"
If player already exists	Move(2,2)	Should throw an exception if the other player or the player himself has made a move there
If game over	Move(2,2)	"IllegalStateException"

Testing getTurn()	Input Value	Expected Value
Player 2 gets turn after player 1	getTurn()	pass
Game is over	getTurn()	"IllegalStateException"
Player asks for the turn when another player is taking his turn	getTurn()	"IllegalStateException"

Testing isGameOver()	Input Value	Expected Value
Player 1 wins	isGameOver()	True
Player 2 wins	isGameOver()	False
Game is Tied	isGameOver()	Null

Testing getWinner()	Input Value	Expected Value
Player 1 wins	getWinner()	X | O | O X | X | O X | O | null Player 1 Wins
Player 1 wins and all markings are at different array indices	getWinner()	"IllegalStateException"
Player 1 wins and all markings are at same indices, but player 2 is returned	getWinner()	"Exception"

Testing getMarkAt(row,column)	Input Value	Expected Value
A player has marked X at row 2 column 0	getMarkAt(2,0)	X
Negative row value	getMarkAt(-2,0)	"IllegalArguementException"
Negative Column Value	getMarkAt(0,-2)	"IllegalArguementException"
Out of bound value	getMarkAt(8,9)	"IndexOutOfBound"

Testing getBoard()	Input Value	Expected value
Returns state of the board	getBoard()	X | O | O X | X | O X | O | null
Correct state should be returned with correct indices	getBoard()	"Exception"