Method To Be Tested	Test Description
testIsLegalVerticalNonCaptureMove	● Test to make sure that it is conditional to cover all the branches. ● This tests the over all the conditional branches which include ● The backwards, and forward move to make sure they are captured ● Tests in JUnit show whether the forward and backward move are able to capture vertically when horizontally moved ● The dorward move works however one of the backward moved fails when compared to the expected value
testIsLegalHorizontalNonCaptureMove	● Test to make sure that the a legal horizontal non capture is moved through all the test cases ● This test will be checked over the conditional branches of the forward (1), (2), move and backward move and the max allowed step ● This test will be checked
testIsLegalDiagnalNonCaptureMove	● This method must be checked over all the conditional branches The conditional branches include the move for forward and backward when the max allowed steps, and one step ● If it is a non capture diagonal movement is made it must check the possible backward and forward conditions
testIsLegalNonCaptureMoveForQueen	● This method must be checked over all the conditional branches of queen if a legal non capture movement is made ● The conditional branches include forward move horizontal

	● As Well as backward move that is horizontal ● Another branch is the forward move that is vertical ● As Well as backward move that is vertical ● Another conditional branch involves a diagonal move in the forward directional ● As Well as the diagonal movement in the backward direction
testIsLegalCaptureMoveForQueen	● This test is made to ensure that if a legal capture movement for a queen piece than the queen is able to capture ● This method must be able to check all the conditional branches ● The conditional branches include if a legal move is made and is in the forward diagonal directional if it is added there and the piece that it is attacking is removed ● The conditional branch includes if a legal move is made and is in the backward diagonal direction it check to make sure the queen is added there and the pieced attacked is removed ● Another conditional branch that must be checked is if the queen makes a legal capture move in the backward diagonal direction it is made ○ As Well as a legal capture move made by the queen in the backward diagonal move ● Another conditional branch is if the queen can make a legal clause in the forward vertical direction ○ Another conditional branch is if the queen can make a legal

	clause in the backward vertical direction
testIsLegalNonCaptureMoveForRook	● Another test that must be made is to ensure that the rook is able to make a legal non capture move by the rook over the conditional branches ● One of the conditional branches is a forward move in the horizontal directions ● As Well as a horizontal move in the backward direction condition is met when it is a legal non capture move ● Another condition that must be checked for the rook is if it is able to move backward in the vertical direction ● As Well as if the rook is able to move vertically in the backward direction
testIsLegalCaptureMoveForRook	● This test is to check all the conditions of the rooks movement if a capture movement is made and it is legal of the rook ● This test is in place to check if the rook is able to move backward in the horizontal direction and capture a piece ● This test is in place to check if the rook is able to move forward in the vertical direction and capture a piece ● This test is in place to check if the rook is able to move forward in the vertical direction and capture a piece ● This test is in place to check if the rook is able to move backward in the vertical direction and capture a piece
testIsLegalNonCaptureMoveForBishop	● This is the test must ensure that if a legal move is made by the Bishop note

	regarding a capture then it is able to move ● The conditional branches that must be checked include a move made by the bishop in the in a straight diagonal movement ● As well as the conditional that there must be no piece on a square between the bishop's starting and terminating spaces. ● Another conditional that must be checked is a legal move in the backward diagonal direction ● Another conditional that must be checked is a legal move in the forward diagonal direction
testIsLegalCaptureMoveForBishop	● The test is to check all the condition of a Bishops move if it is a legal move not involving a capture for a bishop ● The conditions include A bishop can move any number of spaces in a straight diagonal. ● Another condition is checking that there must be no piece on a square between the bishop's starting and terminating spaces.
testIsLegalNonCaptureMoveForKing	● The test is to check all the condition of a King move if it is a legal move not involving a capture for a king ● The conditions include checking if When a king is threatened with capture (but can protect himself or escape), it's called check. ● Test the condition of If a king is in check, then the player must make a move that eliminates the threat of capture and cannot leave the king in check.

testifPieceIsInPathOfMovement	● This test is in place to ensure that if a piece is in the path of movement it maintains the conditions described in European chess piece ● One movement that must be checked is if it is a pawn and it is in the way the piece of queen, king,knight are able to move over it ● Another condition that must be checked is if the path of movement is blocked for a piece due such as the kind who can not move more than one square or if they are on the same side
testIsLegalCaptureMoveForKing	● The test is to check all the condition of a King move if it is a legal move involving a capture for a king ● The conditions include
testCastlingKing	● This test must be checked ovr all the conditions permitted in European king chess piece ● The condition of if the king is moved two spaces to the left or right ● It must also be checked that the castle move is only legal if the king has never moved ● Another condition that must be checked is if there is a rook belonging to the same player in the corner the king is moving toward and the rook has never moved ● as well as checking the condition of the squares between the king and the rook are threatened inorder to check if the castling move can be made ● Neither king nor rook involved in castling may have moved from the original position; ● There must be no pieces between the

	king and the rook
testIsLegalNonCaptureMoveForKnight	● The test is to check all the condition of a knight move if it is a legal move not involving a capture for a knight ● The conditions include making sure that the Knight can move one square along any direction ● The conditions include making sure that if it is a legal move for a knight not involving capture it can move to the horizontal and vertical forward ● As Well as testing the condition horizontally and vertically downward that the knight is able to move ● And was as testing the condition of whether it can move backward horizontal and vertically upward ● And the condition of whether it can move horizontally and backward downward as well
testIsLegalCaptureMoveForKnight	● The test is to check all the condition of a knight move if it is a legal move involving a capture for a knight piece ● The conditions involve checking if a piece is captured when horizontally and vertically downward that the rook is able to move ● And was as testing the condition of whether it can move backward horizontal and vertically upward ● And the condition of whether it can move horizontally and backward downward as well and be able to capture a piece
testIsLegalNonCaptureMoveForPawn	● The test is to check all the condition of a paul move if it is a legal move not involving a capture for a pawn ● The conditions include making sure

	that they can move forward one square, if that square is unoccupied. ● Another condition includes that if it has not yet moved, the pawn has the option of moving two squares forward provided both squares in front of the pawn are unoccupied. ● Another condition includes that check that a pawn cannot move backward. ● Another condition includes checking that if the pawn lands on the last row on the opponent's side of the board, the pawn can be upgraded to another piece.
testIsLegalCaptureMoveForPawn	● The test is to check all the condition of a paul move if it is a legal move involving a capture for a pawn ● The conditions include making sure that the pawn can move forward one square, if that square is unoccupied and capture another piece ● Another condition includes that check that a pawn cannot move backward and capture a piece ● Another condition includes checking that if the pawn lands on last row on the opponent's side of the board, the pawn can be upgraded to another piece