**HIRARCHY DIAGRAM**

**ChessBoard**

private:

ChessPieces \*Board[8][8];

public:

ChessBoard();

void Move(char color);

void Save\_To\_File();

void Load\_From\_File();

void Start();

void Show();

int Menu();

void Instructions();

void operator=(const int);

~ChessBoard();

**ChessPieces**

protected:

char color;

public:

ChessPieces(char clr);

virtual char Get\_Piece\_Color()=0;

virtual char Get\_Piece\_Symbol()=0;

virtual bool MovePossible(ChessPieces \*Board[8][8], int Src\_Row, int Src\_Col, int Dest\_Row, int Dest\_Col)=0;

~ChessPieces();

**Rook**

private:

char Symbol;

public:

Rook(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);

**Knight**

private:

char Symbol;

public:

Knight(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);

**Pawn**

private:

char Symbol;

public:

Pawn(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);

**Bishop**

private:

char Symbol;

public:

Bishop(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);

**King**

private:

char Symbol;

public:

King(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);

**Queen**

private:

char Symbol;

public:

Queen(char clr);

char Get\_Piece\_Color();

char Get\_Piece\_Symbol();

bool MovePossible(ChessPieces \*Board[8][8],int Src\_Row, int Src\_Col,int Dest\_Row, int Dest\_Col);