# Design Overview for <<insert name of program here>>

Name:

Student ID:

# Summary of Program

Essentially in short my program want to simulate a chess game , which uses polymorphism in design to move pieces. It should use a lot of what we learnt in Object Oriented Programming

# Required Roles

Describe each of the classes and interfaces you will create using the following table (one per record).

Table 1: <<role name>> details

|  |  |  |
| --- | --- | --- |
| Responsibility | Type Details | Notes |
| Chess Game Class | Represents the overall chess game | Main method (Main) for game start and user input. Interacts with the Board Class |
| Board Class | Represents the chessboard and manages the game states. | Methods for initializing , displaying and validating moves as well as updating board state. Uses an array for piece positions |
| Board Initializer | Define the contract for initializing the chessboard. | Interface with initialize method. Implemented by ConsoleBoardInitialize for console-based initialization. |
| ConsoleBoardInitializer | Implements BoardInitalizer for console-based initialization | Provides an implementation for initializing the chessboard for console-based games. Not explicated used but illustrative |
| ConsoleMoveValidator | Validate user moves for console-based games. | Methods for validating user input regarding chess moves. Used by the Board class to ensure correct move format and board boundaries |
| ConsoleChesDisplay | Displays the chessboard for console based games. | Responsible for rendering the current state of the chessboard to the console. Used by the Board class to display the game state |

Table 2: <<enumeration name>> details

|  |  |
| --- | --- |
| Value | Notes |
| Pawn | Represents the pawn chess piece |
| Rook | Represents the rook chess piece |
| Knight | Represents the knight chess piece |
| Bishop | Represents the bishop chess piece |
| Queen | Represents the queen chess piece |
| King | Represents the king chesspiece |

…