**Advance Programming Techniques Assignment 2 Report:**

In our Qwirkle game implementation we have done:

* Created ADTs for the LinkedList, Node, Tile, Board, Game, Bag, Player.
* In our LinkedList we provided functions to implement in the implementing files. These functions are:
* size function to get the size of the LinkedList
* clear function to clear the LinkedList
* get function to get the tiles of the LinkedList
* Add front to add the tiles to the front of the list
* add back to add the tiles to the back of the list
* add at to add the tiles at any specific index of the list
* delete at to delete any tiles from a specific index of the list to provide for constant time instead of the traditional linear time which the LinkedList without this method would grab nodes through linear time.
* Contains method which allows to get the size of the list of tiles
* And delete data function deletes a specific tile from the Linked List
* In terms of implementing the bag of tiles we used a 2D array to iterate through both the colours and shapes of the tile and adding it to the back of the LinkedList and add the size of the bag size by 2 to make sure all pieces of Qwirkle is in the bag twice.
* In terms of implementing the board we implemented it by using a 2D array of pointers to tiles and iterating through the array by for loop for the rows and cols of the board. The efficiency of this is that it allows for quick access to the pointers of tiles into the board the downside to this is that it is iterating through the rows and cols which may be inefficient
* In terms of the look of the board we opted to use nested for loops to iterate through the rows and cols and input the empty spaces which would create the foundation for our pointers to replace thus simulating the gameplay of Qwirkle in the board. In terms of efficiency it is not very efficient as for loops are used to iterate through all rows and cols to print the empty spaces onto the board with lines and dashes to represent the board and ASCII characters to represent the rows and the integers to represent the cols.
* For our gameplay we used separate class to house the implementation of our gameplay functions in order to better facilitate the styling code of our program we provided functions in the ADT of game:
* The getBoard, getPlayer1 and getPlayer2, getTileBag function to be used in other functions in order to simulate gameplay
* replaceTile function to replace tiles in players hand
* add tile to board function to add scores to the current player during gameplay