// public class q2a\_24062219 extends q2b\_24062219{

// public static void main(String[] args) {

// q2b\_24062219 square = new q2b\_24062219();

// int n = square.execVerify(args);

// int[][] execSquared = square.execSquare(n);

// square.execPrint(execSquared, n);

// }

// }

// public class q2b\_24062219 extends q2a\_24062219{

THE PROBLEM AND THE REQUIREMENTS THAT WERE SET OUT.

THE PROBLEMS WE FACED IN MAKING OUR SOLUTION TO THE ORIGINAL PROBLEM.

OUR LEARNING OUTCOMES (links with what the professors wanted us to learn).

OUR FINAL SOLUTION TO THE PROBLEM ITSELF.

2. Write a report that explains what you’ve done and how you did it. Your report should have four sections:

1. a brief overview of the assignment and your goals in completing it.
2. a description of your solution design, including any assumptions made, algorithms used, etc.
3. a discussion of your completed solution to the assignment, including the scope of solution, quality of solution, interesting results, difficulties overcome, enhancements delivered, etc.,
4. a discussion of your software test methodology. How did you ensure that your solution does what it's meant to do.

Goals: Create a puzzle game that features a magic square.

Develop a Object-Oriented program that has input and output functionality and that is event driven. 2. Show fluency in selecting and using basic components in the Java language.

Any n x n magic

square (where n is an odd integer) consists of an n x n matrix whose elements contain the numbers 1,2,3,….,n2 such that the sum of each row, column and diagonal is equal to !(!!#$) & . For example, the following magix square for n = 3, with the sum of each row, column and diagonal being '('!#$) & = 15:

Write a command line application

A fully functional game prototype

\* Documentation detailing the design decisions and technical

implementation

\* Testing plans and results to demonstrate the game's

effectiveness

**GOALS**

Create an engaging event driven object-oriented java puzzle game that challenges players to solve a magic square using input and output functionality. This program has to be a fully functional prototype of a game that shows fluency in selecting and using basic components in Java.

The aim of the program was to make a comprehensive and understandable game whereby the user can chose the size of the solvable array to make and shuffle a magic square matrix and allow the user to use inputs and outputs to remake the array in to a magic square.