# **Sets Report**

# Design

## Introduction

For this assignment it would have been difficult to compare lists to see which integers match the random lottery generation that the user has entered. As the matching integers could be placed anywhere within the list meaning each item must be checked systematically. Therefore sets had to be used instead as the intersection of two sets could be used to check for a winner instead. This is more efficient as the placement in the set is less important. Thus it is also a lot easier to have multiple players in the lottery.

## Requirements

***Functional:***

***R1.*** *Allow a player to select 6 lottery numbers, validate them and store them in a Set.*

***R2.*** *Run the lottery, i.e. generate 6 numbers between 1 & LOTTERY\_MAX using a random number generator and place these in a set too.*

***R3.*** *Find out if the user has won, and if so, how much they have won.*

***R4.*** *Enable the user to define the range of lottery numbers to use, e.g. a valid range might run from 1 to 11 for a small lottery which would give each player a greater chance of winning, your program should be able to run with the range specified by the user.*

***R5.*** *Extend your program to cater for more than one player with each lottery run. This should be a variable number of players specified by the user or a fixed number of players.*

***R6.*** *Use a custom “MySet” classto handle the complexity of the set operations, including an intersection method.*

I was able to tackle and successfully complete all of these requirements however I would have liked to have had several weeks as well.

## Pseudocode

### intersection method

*Sets set of integers “tmpset” to “set”*

*“tmpset” retains all of its set and passes intersect of “set”*

*Returns “tmpset”*

### runLottery method

*Creates a for loop that runs 6 times*

*Creates a while loop*

*Creates a new random called “r”*

*Declares an integer “rnum” and sets it to a random integer between*

*lotteryMax and lotteryMin*

*If adding “rnum” is true*

*Break loop*

*Calls “winCheck” method*

## Class diagram

Creates object

|  |
| --- |
| **LottoNum** |
| * userSet: ArrayList * random: MySet * lotteryMax: int * lotteryMin: int |
| * userInput() * runLottery() * winCheck() |

|  |
| --- |
| **MySet** |
| * set: HashSet<Integer> |
| * add(int n) * intersection() * size() * clear() |

|  |
| --- |
| **Main** |
| * main() |

Creates object

## AC12001 – Test Plan

Name: …Isaac Lowry…………………………………………………………………………..

Matric number: …170025555 …………………………………………………….……….

Lab Title: …AC12001 assignment 2: Sets………………………………..……….

Test number/date/version: 12/02/18 ……………………………………….……..

Test Notes: …Tests completed manually ……………………………

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected result** | **Worked?** |
| Player count | a | Outputs “Wrong input” error | Y |
| Player count | 0 | Outputs “Wrong input” error | Y |
| Player Count | 2 | Outputs message asking for lower and upper boundaries | Y |
| Range | 2  4 | Outputs “Wrong input” error | Y |
| Range | 2  10 | Outputs message asking for lottery numbers | Y |
| Range | a | Outputs “Wrong input” error | Y |
| Range | 1 10 | Outputs message asking for lottery numbers | Y |
| User Lottery Numbers | 2 2 2 4 7 6 | Outputs message asking for lottery numbers again | Y |
| User Lottery Numbers | A 1 2 3 4 5 | Outputs “Error” error | Y |
| User Lottery Numbers | 1 2 3 4 5 6 | Outputs whether or not user has won | Y |
| User Lottery Numbers | 1 2 3 4 5 | Outputs message asking for lottery numbers again | Y |
| User Lottery Numbers | 1 2 3 4 5 6 7 | Outputs message asking for lottery numbers again | Y |
| User Lottery Numbers | 123456 | Outputs message asking for lottery numbers again | Y |
| User Lottery Numbers | 1,2,3,4,5,6 | Outputs message asking for lottery numbers again | Y |
| User Lottery Numbers | -9 8 7 6 5 4 | Outputs whether or not user has won | Y |

# Self-evaluation

I found this assignment relatively challenging as I had to learn the concept of sets. I found completing the assignment to be the most enjoyable as a lot of exception handling was required making the creation of the lottery relatively tricky. I was proud to have achieved all the requirements as well as the optional requirement of having a custom set class. However I found extending the program to multiple weeks to be too hard due to my lack of programming experience and poor time management.