

Assignment 1

Submission: via Blackboard

Points: 50

Objectives:

1. Follow software specifications provided to design and implement a program
2. Design your program using a flowchart and identify list of modules/methods/functions
3. Conduct design review (self)
4. Implement the program with appropriate JavaDoc Documentation
5. Test case generation and testing

Program Specification:

An $n \times n$ matrix that is filled with the whole numbers 1, 2, 3, ... n^2 is a *magic square* if the sum of the elements in each row, in each column, and in the two diagonals is the same value.

Here is a magic square where $n = 3$:

8	1	6
3	5	7
4	9	2

Write a program that reads n^2 numbers from standard input and tests whether they form a magic square when put into matrix form. The value of n is **NOT** an input to the program; n must be determined from the number of inputs.

For example, the input that creates the example matrix above is: 8 1 6 3 5 7 4 9 2

The output is a single word, "true" if the input produces a magic square, "false" otherwise. Your program may assume that each input token is a whole number.

The program must verify:

1. The proper number of input values was provided.
2. Each of the numbers between 1 and n^2 occurs exactly once in the input.
3. When the numbers are arranged in a matrix, the sum of the
 - rows,
 - columns,
 - and diagonalsmust be the same value.

Step 1: Design (10 points)

Design a Magic Squares program as per specifications provided above. Create a flow chart and identify a list of modules/functions/methods.

Step 2: Design Review (Self) (5 points)

Review your design using the Design review checklist provided. Mark the checklist with your findings and make necessary changes.

Preliminary Design Review

- i. Traceability
 - Did you ensure traceability of the design back to the systems specification and statement of requirements?
 - Is a scheme used for naming of modules, data, and interfaces?
 - Are all modules, data, and interfaces uniquely identified?
- ii. Consistency
 - Is the data structure consistent with the information domain?
 - Is the data structure consistent with software requirements?
 - Is a standard design representation used?
 - Is a standard data usage representation used?
- iii. Completeness
 - Are software requirements reflected in the software architecture?
 - Are all referenced data defined?
 - Are all defined data used?
 - Are all referenced modules defined?
 - Are all defined modules used?
 - Are interfaces defined for modules and external elements?
 - Has maintainability been considered?
- iv. Efficiency
 - Are data grouped for efficient processing?
 - Are storage requirements allocated to design?
 - Is effective modularity achieved? Are modules functionally independent?

Step 3: Software size estimation (7 points)

Estimate the program size. Use the following informal procedure for the estimation and mark the estimation table below.

Informal Size Estimating Procedure:

1. Study the requirements.
2. Sketch out a crude design.
3. Decompose the design into “estimable” chunks.
4. Make a size estimate for each chunk, using a combination of:
 - * visualization
 - * recollection of similar chunks that you’ve previously written
 - * intuition
5. Add the sizes of the individual chunks to get a total.

SIZE ESTIMATION TABLE

Module Description	Estimated Size
TOTAL ESTIMATED SIZE	

Step 4: Implementation with JavaDoc Comments (20 points)

Implement the program in java. Name your program MagicSquare.java. Make sure you provide JavaDoc comments and generate documentation files (html files).

Step 5: Testing (5 points)

Identify 4 test cases; one where each of criteria 1, 2, 3 provided in the program specification fail, and one where they all pass. Test your program for all 4 test cases. Provide the outcome of your program for the 4 test cases.

Test Case Name	Input	Output
1:		
2:		
3:		
4:		

Step 6: Actual Software Size (3 points)

Document and examine the Estimated vs. Actual software size:

Module Description	Estimated Size	Actual Size
TOTAL SIZE		

SUBMISSION:

Submit this document along with your source code (java program) and JavaDoc documentation generated (html files) archived as ASUUserName-Assignment1.zip via Blackboard.