Vibhor Agarwal
Section B
2020349

Advanced Programming
Assignment_3 Writeup

Task 1

- OOPS, principles helped me in this task as, since all the different types of matrices are just the type of a common matrix class, we can extend the functionality of the matrix class to specialised matrix types like diagonal matrix, symmetric matrices and so on. Also using OOPS principles we can store each different type of matrix object in a list that stores the matrix and can override the special functionalities in child classes like transpose is the same as a matrix for inverse matrices etc.

Task 2

- Using OOPS principles we can have the required matrix types in our inheritance tree and when we need to create a matrix of a fixed matrix type we can simply create it using predefined values from our side. Also, OOPS principles make it easier to store different types of matrices in a single list for easier access and retrieval.

Task 3

- Using OOPS Principles we can have a check on the matrix type a matrix object is stored. If the user changes the element in such a way that the type of the matrix stored is changed then we would get an error and it would act as a protection for the same. Also, we can check the type of an object matrix without even creating the object of that matrix by defining the findMatrixTypeStored function in another manager class which we instantiate already.

Task 4

- With the help of the OOPS principles, we can store different types of matrices in a single matrix list using method overriding and polymorphism. In this way, the matrix retrieval becomes easier as all the matrices are located in a single list so we can simply traverse through the list and display all the matrix type labels for the requested objects

Task 5

- In this task OOPS principle helped me as, if we know if out of two operand matrices if a matrix is a null matrix then the addition and subtraction is the other matrix itself and multiplication result is always the null matrix, so we don’t need to do the calculation again and again for null matrix.
- Also for the identity matrix, the result of the multiplication is simply the other matrix itself, so in this case, we don’t need to do all the calculations for getting the result. We can simply override the multiplication method for the identity matrix and show the other matrix as result without any calculation.

Task 6

- For element-wise addition and subtraction with null matrix, the result is the other matrix itself. Also for multiplication with null matrix, the result is null matrix itself. For element-wise division with null matrix, we simply throw an error as we cannot divide by zero.

Task 7

- For Transpose, we know that symmetric matrices and skew-symmetric matrices have special properties. For symmetric matrices transpose is the same as the matrix itself so we just override the transpose method in the symmetric matrix and just display the matrix without any calculation. And for the skew-symmetric matrix, we just multiply the matrix by -1 and show the result.

- Since identity and null matrix are also symmetric matrices we do the same for these matrices also.
- We also do the same for Diagonal matrices as they are also symmetric matrices

Task 8

- For the identity matrix, we know that the inverse is the same as the identity matrix itself so we just print the identity matrix.
- For the null matrix, the inverse does not exist so we just throw an error without any calculation.
- For singular matrix, we also throw an error without any calculation.
- For rectangular matrices also the inverse does not exist so we simply throw an error.

Task 9

- Here for the null matrix the mean row-wise, column-wise and for all elements is always zero. So we just print zero without any further calculations.

Task 10

- For diagonal matrices, the determinant can simply be calculated by just multiplying the diagonal elements
- For the identity matrix, the determinant is always 1
- For null matrix, the determinant is always 0
- For the rectangular matrix inverse does not exist so we simply throw an error.
- For the diagonal matrix, the determinant is simply the product of the diagonal elements.

Task 11

- If a scalar is being multiplied with a null matrix then we simply output the null matrix without any calculation
- In the case of the Identity matrix, we simply multiply the diagonal 1's with scalar and show the result
- In the case of the diagonal matrix, we simply multiply the diagonal elements with the singleton matrix as a scalar and show the result.

Task 12

- For Symmetric matrices, A+ At is same as 2*A. So we multiply each element of the matrix with 2 and show the output
- For Skew Symmetric matrices A+ At is a null matrix. We simply output a null matrix in this case without any further calculations.
- For null matrix A+At is same as null matrix itself

Task 13

- For the null matrix, the eigenvalues are always zero. Also for a null matrix, any matrix of the same dimension is the eigenvector. So we just output any random matrix of the same dimensions.

- For identity matrix the eigenvalues are always 1 . Also for a null matrix, any matrix of the same dimension is the eigenvector. So we just output any random matrix of the same dimensions.

Task 14

- Here if in AX=B, if A is an identity matrix the solution matrix is simply the B matrix itself so we just output it without any calculations.
- Also if A is a null matrix and B is not a null matrix then we simply throw an error.
- Also if A is not a null matrix but B is a null column matrix then we simply output B as the result.

Task 15

- With the help of the OOPS principle we can store the objects of different types but belonging to the same parent in a parent type Array List using method overriding and polymorphism. This makes the searching of the objects in the list very easy and we can simply traverse through the array list to find the required matrix types.