# 1.7 Abstraction

## In my video, I will show

- Make P(3x4) and P(4x3) multiplication
  - M = PQ
- Inverse Matrix matA(4x4)
  -Copy Matrix Code and make a class “Mat”
  - Call Mat.inverse(matA)
  - X1 = A*inverse A
  - X2 = inverse A * A
- Y1=A*B
- Y2=inverse A*Y1
- Y3=Y2-B

- Make an exception code.
  -Try catch

# Abstraction


- Watch the new videos.
- https://www.youtube.com/watch?v=Mm06BuD3PlY&list=PLLAZ4kZ9dFpPpdR_9IQBUDLjYalvdrGGb&index=26
- Write and run all of the programs

in the videos

- Abstraction = Reuse programs with only inputs and outputs

- Copy “Matrix” and make your class “Mat”
- https://github.com/rchen8/algorithms/blob/master/Matrix.java

# Inverse Matrix

$$A \bullet A^{-1} = \begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 6 \\ 1 & 3 & 0 \end{pmatrix} \bullet \left[ \frac{1}{37} \begin{pmatrix} -18 & 15 & 13 \\ 6 & -5 & 8 \\ 11 & -3 & -10 \end{pmatrix} \right]$$

$$= \frac{1}{37} \begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 6 \\ 1 & 3 & 0 \end{pmatrix} \bullet \begin{pmatrix} -18 & 15 & 13 \\ 6 & -5 & 8 \\ 11 & -3 & -10 \end{pmatrix}$$

$$= \frac{1}{37} \begin{pmatrix} -36+18+55 & 30-15-15 & 26+24-50 \\ -72+6+66 & 60-5-18 & 52+8-60 \\ -18+18+0 & 15-15+0 & 13+24-0 \end{pmatrix}$$

$$= \frac{1}{37} \begin{pmatrix} 37 & 0 & 0 \\ 0 & 37 & 0 \\ 0 & 0 & 37 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Change every “private”

public

```
import java.util.Arrays;

public class Matrix {

        private static double determinant(double[][] matrix) {
                if (matrix.length != matrix[0].length)
                        throw new IllegalStateException("invalid dimensions");

                if (matrix.length == 2)
                        return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0];

                double det = 0;
                for (int i = 0; i < matrix[0].length; i++)
                        det += Math.pow(-1, i) * matrix[0][i]
                                        * determinant(minor(matrix, 0, i));
                return det;
        }

        private static double[][] inverse(double[][] matrix) {
                double[][] inverse = new double[matrix.length][matrix.length];

                // minors and cofactors
```