ENSF 614 – Fall 2021

Lab 4 – Tuesday, October 19

Student Name: Aastha Patel and Bhavyai Gupta

Submission date: October 19, 2021

# Exercise A – Source file MyArray.cpp

/\*

 \* File Name:               MyArray.cpp

 \* Course:                  ENSF 614 - Fall 2021

 \* Lab # and Assignment #:  Lab 4 Exercise A

 \* Lab section:             B01

 \* Completed by:            Bhavyai Gupta, Aastha Patel

 \* Submission Date:         NA

 \*/

#include "MyArray.h"

MyArray::MyArray()

{

    sizeM = 0;

    storageM = new EType[sizeM];

}

MyArray::MyArray(const EType \*builtin, int sizeA)

{

    sizeM = sizeA;

    storageM = new EType[sizeM];

    for (int i = 0; i < sizeA; i++)

    {

        storageM[i] = builtin[i];

    }

}

MyArray::MyArray(const MyArray &source)

{

    sizeM = source.size();

    storageM = new EType[sizeM];

    for (int i = 0; i < sizeM; i++)

    {

        storageM[i] = source.storageM[i];

    }

}

MyArray &MyArray::operator=(const MyArray &rhs)

{

    if (this != &rhs)

    {

        delete[] storageM;

        sizeM = rhs.size();

        storageM = new EType[sizeM];

        for (int i = 0; i < sizeM; i++)

        {

            storageM[i] = rhs.storageM[i];

        }

    }

    return \*this;

}

MyArray::~MyArray()

{

    delete[] storageM;

    storageM = nullptr;

}

int MyArray::size() const

{

    return sizeM;

}

EType MyArray::at(int i) const

{

    return storageM[i];

}

void MyArray::set(int i, EType new\_value)

{

    storageM[i] = new\_value;

}

void MyArray::resize(int new\_size)

{

    EType \*temp = new EType[new\_size];

    int limit = (new\_size > sizeM) ? new\_size : sizeM;

    for (int i = 0; i < limit; i++)

    {

        temp[i] = storageM[i];

    }

    delete[] storageM;

    sizeM = new\_size;

    storageM = temp;

}

# Exercise A – Program Output

Text

Description automatically generated

# Exercise B – Source Code of transpose

# Exercise B – Program Output

# Exercise C – Source Code of print\_from\_binary