



# Computer Science 3A

## Practical Assignment 1

15 February 2023

Time: 15 February 2023 — 17h00

Marks: 50

---

Practical assignments must be uploaded to `eve.uj.ac.za` **before** 17h00. Late submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Operations on Images are very common in many Computer Vision and Image Processing areas. In this practical, you must implement a program demonstrating various Image operations. You must use Java as the implementation language, and your program should consist of various classes where you implement your solution.

You are required to implement the following image-related operations:

- Write Bitmap from Image
- Image Scalar Multiplication
- Image Scalar Addition
- Image Transpose

### Image Scalar Addition

Consider a constant  $c$  that must be added to the matrix  $A$ .

$$c + A = c + A_{i,j} \begin{bmatrix} c + a_{1,1} & c + a_{1,2} & \cdots & c + a_{1,n} \\ c + a_{2,1} & c + a_{2,2} & \cdots & c + a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c + a_{m,1} & c + a_{m,2} & \cdots & c + a_{m,n} \end{bmatrix}$$

### Image Scalar Multiplication

Consider a constant  $c$  that must be added to the matrix  $A$ .

$$cA = cA_{i,j} \begin{bmatrix} c \times a_{1,1} & c \times a_{1,2} & \cdots & c \times a_{1,n} \\ c \times a_{2,1} & c \times a_{2,2} & \cdots & c \times a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c \times a_{m,1} & c \times a_{m,2} & \cdots & c \times a_{m,n} \end{bmatrix}$$

## Image Transpose

Consider the Transpose of the Matrix  $A$ :

$$A^T = \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{m,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{m,n} \end{bmatrix}$$

The following files must be submitted to EVE:

1. *studentnumber\_p1.zip*

## Creating Arrays with Generics

Creating an array in a generic object can be difficult. It is possible to do it with Reflection in Java 7, but this is incompatible with other versions of Java and can be problematic when getting the base class type from passed generic parameters.

The easiest way to work with a generic array is to look at the following class:

```
public class MyArray<E> {
    private E[] arr;

    public MyArray(Integer s) {
        genArray(s);
    }

    public void genArray(int s) {
        final Object[] a = (E[]) new Object[s];
        arr = (E[]) a;
    }

    public void setElement(int x, E value) {
        arr[x] = value;
    }

    public E getElement(int x) {
        return arr[x];
    }
}
```

This does not produce very safe code but creates a generic array. You can encounter type erasure problems if the objects are incorrectly handled. It is easily extendible for 2-dimensional arrays. You should not use the above class in your solution, just the concepts presented in the above class.

## Marksheet

- |  |      |
|--|------|
| 1. BitmapConvert: writeBitmapFromArray | [15] |
| 2. Image: instantiateArray             | [5]  |
| 3. ImageOperator: addScalar            | [5]  |
| 4. ImageOperator: multiplyScalar       | [5]  |
| 5. ImageOperator: transpose            | [5]  |
| 6. Compilation and Correct execution   | [15] |