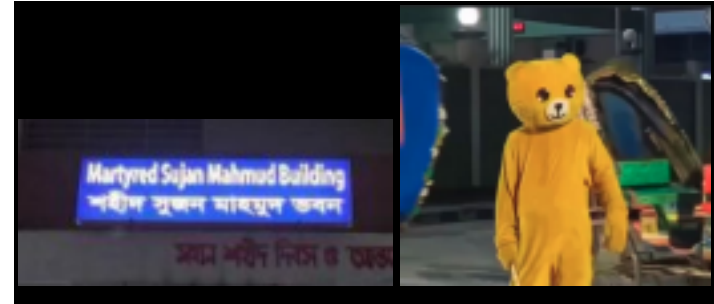




**Welcome back, heroes!**



**Bangladesh 2.0**



## Chapter 3 - Data Types, Variables, and Arrays

| Data Type | Size (bits) | Range   | Example                |
|-----------|-------------|---|------------------------|
| `byte`    | 8           | $-2^7$ to $2^7-1$                                       | `byte b = 100;`        |
| `short`   | 16          | $-2^{15}$ to $2^{15}-1$                                 | `short s = 10000;`     |
| `int`     | 32          | $-2^{31}$ to $2^{31}-1$                                 | `int i = 100000;`      |
| `long`    | 64          | $-2^{63}$ to $2^{63}-1$                                 | `long l = 100000L;`    |
| `float`   | 32          | $\pm 1.4 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$   | `float f = 234.5f;`    |
| `double`  | 64          | $\pm 4.9 \times 10^{-324}$ to $\pm 1.8 \times 10^{308}$ | `double d = 123.4;`    |
| `char`    | 16          | 0 to $2^{16}-1$   | `char c = 'A';`        |
| `boolean` | 1           | true or false   | `boolean bool = true;` |

## Data Type: Characters

Although char is designed to hold Unicode characters, it can also be thought of as an integer type on which you can perform arithmetic operations.

```
// char variables behave like integers.
class CharDemo2 {
    public static void main(String args[]) {
        char ch1;

        ch1 = 'X';
        System.out.println("ch1 contains " + ch1);

        ch1++; // increment ch1
        System.out.println("ch1 is now " + ch1);
    }
}
```

The output generated by this program is shown here:

```
ch1 contains X
ch1 is now Y
```

## Data Type: Boolean

```
// Demonstrate boolean values.
class BoolTest {
    public static void main(String args[]) {
        boolean b;

        b = false;
        System.out.println("b is " + b);
        b = true;
        System.out.println("b is " + b);

        // a boolean value can control the if statement
        if(b) System.out.println("This is executed.");

        b = false;
        if(b) System.out.println("This is not executed.");

        // outcome of a relational operator is a boolean value
        System.out.println("10 > 9 is " + (10 > 9));
    }
}
```

The output generated by this program is shown here:

```
b is false
b is true
This is executed.
10 > 9 is true
```

## Variables: Dynamic Initialization

```
// Demonstrate dynamic initialization.
class DynInit {
    public static void main(String args[]) {
        double a = 3.0, b = 4.0;

        // c is dynamically initialized
        double c = Math.sqrt(a * a + b * b);

        System.out.println("Hypotenuse is " + c);
    }
}
```

## Scope

```

class Scope {
    public static void main(String args[]) {
        int x; // known to all code within main

        x = 10;
        if (x == 10) { // start new scope
            int y = 20; // known only to this block

            // x and y both known here.
            System.out.println("x and y: " + x + " " + y);
            x = y * 2;
        }

        // y = 100; // Error! y not known here

        // x is still known here.
        System.out.println("x is " + x);
    }
}

```

## Lifetime of a Variable

```
// Demonstrate lifetime of a variable.
class LifeTime {
    public static void main(String args[]) {
        int x;

        for (x = 0; x < 3; x++) {
            int y = -1; // y is initialized each time block is entered
            System.out.println("y is: " + y); // this always prints -1
            y = 100;
            System.out.println("y is now: " + y);
        }
    }
}
```

```
// This fragment is wrong!
count = 100; // oops! cannot use count before it is declared!
int count;
```

```
int bar = 1;
{
    // creates a new scope
    int bar = 2; // Compile-time error - bar already defined!
}
```

## Arrays: 1 Dimensional



```
public static void main(String args[]) {
    int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    System.out.println("April has " + month_days[3] + " days.");
}
```

`int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };` declares and initializes the array in a single step.

`System.out.println("April has " + month_days[3] + " days.");` accesses the fourth element (index 3) of

```
public static void main(String args[]) {
    int month_days[];
    month_days = new int[12];
}
```

`month_days = new int[12];` allocates memory for 12 integers in the array `month_days`.

## Arrays: 2 Dimensional



Each inner set of curly braces `{ }` represents a row in the two-dimensional array.

## Arrays: 2 Dimensional



11

## Arrays: Declaration Styles





Chapter 3

## Problem: Matrix Addition

Write a Java program that performs the following tasks:



1. Prompt the user to enter the dimensions (number of rows and columns) of two matrices.
2. Read and store the elements of both matrices.
3. Add the two matrices and store the result in a third matrix.
4. Print the resulting matrix.

**That's All!**

