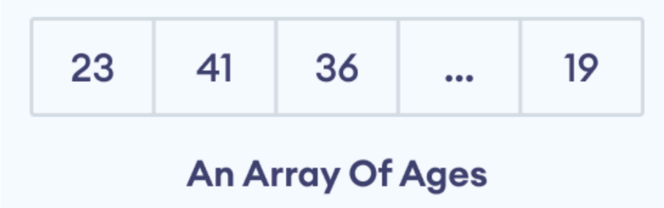
Programming Concept Practiced

1. **Java Arrays -** So far we have used simple variables to store single data with arrays we can store multiple data. Suppose we want to store the ages of all students in a class and say there are 40 students, we can define 40 variables which will make it a lengthy and tedious task. Instead, we can create a single array and store all the ages as in the figure below.



23	41	36	...	19

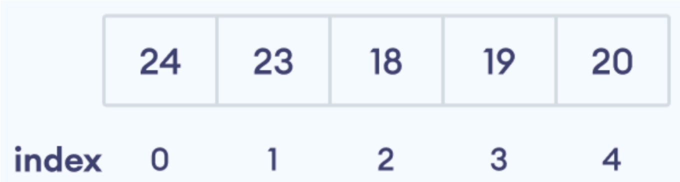
An Array Of Ages

2. **Create an Array -** An array stores values of a particular data type, has a name like any variable, is indicated using square brackets ***[ ]***, and can be initialized using curly braces ***{ }*** as in the example shown below for an array of ages

```
int[] age = {24, 23, 18, 19, 20};
```

   a. ***age*** - the name of the array
   b. ***[ ]*** - represents it as an array
   c. ***int*** - represents that the array can only include integer elements.
   d. ***{24, 23, 18, 19, 20}*** - elements of the array

3. **Access Array Elements -** In Java, each element of an array is associated with a number called **array index**. The array index specifies the position of the element inside the array as shown in the figure below



	24	23	18	19	20
index	0	1	2	3	4

   a. An array index starts with 0, so the first element of the array will always be at position 0.
   b. The following code prints all the elements of the array

```
int[] age = {24, 23, 18, 19, 20};
System.out.println(“Age at first element is: “ + age[0]);
System.out.println(“Age at second element is: “ + age[1]);
System.out.println(“Age at third element is: “ + age[2]);
System.out.println(“Age at fourth element is: “ + age[3]);
System.out.println(“Finally Age at fifth element is: “ + age[4]);
```

4. **Accessing Array Elements Using *for* Loop -** It's efficient to access array elements using loops as shown below

```
Java
// Creating an array of ages
int[] age = {24, 23, 18, 19, 20};

// Printing every element of age using for loop
for (int index = 0; index < 5; ++index) {
   System.out.println(“Age at element " + (index+1) + " is: “ + age[index]);
}
```

5. **Modifying Array Elements -** We can modify or change the value of array elements by simply accessing and assigning value. Suppose we want to increment age by 1 year, we can do the following

```
int[] age = {24, 23, 18, 19, 20};
age[0] = 25; // simply assigning a new value
age[1] = age[1] + 1; // Accessing the element and increment by 1
Age[2]++; // Accessing the element and using the increment operator
```

This can be done efficiently using a ***for*** loop.

```
Java
// Creating an array of ages
int[] age = {24, 23, 18, 19, 20};

// Incerment age by 1 year and Printing the age using for loop
for (int index = 0; index < 5; ++index) {
   age[index]++;
   System.out.println(“Age at element " + (index+1) + " is: “ + age[index]);
}
```

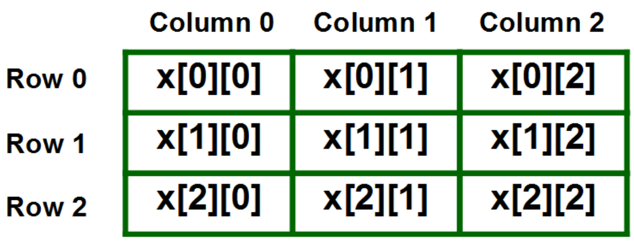
6. **Size of the Array -** Java has an inbuilt property ***length*** to find the size of the array. The syntax is

```
int[] age = {24, 23, 18, 19, 20};
System.out.println("The size of the array is ” +age.length);
```

7. **Java Multi-Dimensional Arrays -** A multidimensional array can be defined as an array of arrays. Data in multidimensional arrays are stored in tabular form up to the nth Dimension.
   a. Like in the single-dimensional array we have seen so far, the multi-dimensional array is defined with the data type, array name, and the number of dimensions indicated by square brackets ***[ ]*** as in `int[][][] 3DArray = new int[3][3][3];`

b. **Size of Multidimensional Arrays:** The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions. In the above example, the size of the 3DArray variable is 3*3*3 = 27

c. **Access elements of a multi-dimensional array -** The multi-dimensional array is accessed through a for loop and each additional dimension there is an inner for loop. The figure below depicts the 3D array



	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

```
Java
// Create a 3D array (matrix) of integers of size 3 x 3 x 3
int[][][] matrix = new int[3][3][3];

// Initialize it with some values
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < 3; k++) {
            matrix[i][j][k] = i + j + k;
        }
    }
}

// Display the 3D array
System.out.println("\n3D Array:");
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < 3; k++) {
            System.out.print(matrix[i][j][k] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
```

8. **Two – Dimensional Array (2D-Array) -** 2 – Dimensional (2D) array is the simplest form of a multidimensional array. A 2D array can be seen as an array storing multiple 1-D arrays for easier understanding**.** The dimensions of the 2D array are commonly called rows and columns

9. **Java *System.exit()* Method -** This method is used to exit current program by terminating the running Java virtual machine. This method takes a status code.
    a. ***exit(0)*** : Generally used to indicate successful termination.
    b. ***exit(1)*** or ***exit(-1)*** or any other non-zero value – Generally indicates unsuccessful termination.

10. **Java *System.err* message -** System.err is a PrintStream to write to user terminal and works like ***System.out.println*** except it is normally only used to output error texts. Some programs (like Eclipse) will show the output to System.err in red text, to make it more obvious that it is error text. The syntax is

    E.g. => Perform a check for Invalid Grades. In such a case exit and terminate the program. For valid grades Solve the Pass or Fail

```java
import java.util.Scanner;

class StudentResultIndicator {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // get student score from the user
        System.out.println("Enter grade");
        int studentGrade = input.nextInt();

        // condition for invalid score
        if (studentGrade < 0 || studentGrade > 100) {
            System.err.println("Invalid grade.");
            System.exit(0);
        }

        // if studentGrade is 50 or above, display pass and congratulatory
        // message else display failed message
        if (studentGrade >= 50) {
            System.out.println("Congratulations!");
            System.out.println("You passed the examination.");
        } else {
            System.out.println("Sorry.\nYou failed the examination.");
        }
        input.close();
    }
}
```

Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Avoid Hard Coding of variables wherever possible
3. Proper naming conventions for all variables
4. Proper Program Name and Class Name
5. Follow proper indentation
6. Give comments for every step or logical block like a variable declaration or conditional and loop blocks
7. For every user input validate the user input, if invalid, state the error either exit the program or ask user to enter again
8. Use Array ***length*** property while using ***for*** loop

1. **Sample Program 1 -** Create a program to find the sum of all the digits of a number given by a user using an array and display the sum.

   Hint =>

   a. Take the input for a number and validate, if failed state and exit the program
   b. Find the count of digits in the number
   c. Find the digits in the number and save them in an array
   d. Find the sum of the digits of the number and display the sum

```java
// Create SumOfDigit Class to compute the sum of all digits of a number using
// an array
import java.util.Scanner;

class SumOfDigits {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

      // Take input for a number
      System.out.print("Enter a number: ");
      int number = input.nextInt();

      // Validate the user input number, if negative state invalid and exit
      if (number < 0) {
            System.err.println("Invalid Number.");
            System.exit(0);
      }
      // Find the count of digits in the number
      int count = 0;
      int temp = number;
      while (temp > 0) {
            count++;
            temp /= 10;
      }

      // Find the digits in the number and save them in an array
      int[] digits = new int[count];
      for (int i = 0; i < count; i++) {
            digits[i] = number % 10;
            number /= 10;
      }

      // Find the sum of the digits of the number
      int sum = 0;
      for (int i = 0; i < count; i++) {
            sum += digits[i];
      }

      // Display the sum of the digits of the number
      System.out.println("\nSum of Digits: " + sum);

      // Close the Scanner Object
      input.close();
   }
}
```

2. **Sample Program 2 -** Working with Multi-Dimensional Arrays. Write a Java program to create a 2 Dimensional (2D) array (matrix) of integers, initialize it with values, and print the sum of all elements in the matrix

   Hint =>

   a. Take the input for a number of rows and columns
   b. Create a 2D array (matrix) of integers
   c. Take the input for the elements of the matrix
   d. Calculate the sum of all elements in the matrix and display the sum
   e. Also, Display the matrix

Java

```
// Program to create a 2D array, display the elements and calculate the sum of
// the elements of the array
import java.util.Scanner;

class 2DArray {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

      // Declare the 2D Array
      int[][] arr = new int[3][3];

      // Input the elements of the 2D Array
      System.out.println("Enter the elements of the 2D Array: ");
      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            arr[i][j] = input.nextInt();
         }
      }

      // Display the elements of the 2D Array and calculate the sum of the
      // elements of the 2D Array
      int sum = 0;
      System.out.println("The elements of the 2D Array are: ");
      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            System.out.print(arr[i][j] + " ");
            sum += arr[i][j];
         }
         System.out.println();
      }

      // Display the sum of the elements of the 2D Array
      System.out.println("The sum of the elements of the 2D Array is: " + sum);

      // Close the Scanner Object
      input.close();
   }
}
```