

# 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();
    }
}
```

## Level 1 Practice Programs

1. Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.

**Hint =>**

- a. Define an array of 10 integer elements and take user input for the student's age.
  - b. Loop through the array using the length property and for the element of the array check If the age is a negative number print an invalid age and if 18 or above, print The student with the age \_\_\_\_ can vote. Otherwise, print The student with the age \_\_\_\_ cannot vote.
2. Write a program to take user input for 5 numbers and check whether a number is positive, negative, or zero. Further for positive numbers check if the number is even or odd. Finally compare the first and last elements of the array and display if they equal, greater or less

**Hint =>**

- a. Define an integer array of 5 elements and get user input to store in the array.
  - b. Loop through the array using the length If the number is positive, check for even or odd numbers and print accordingly
  - c. If the number is negative, print negative. Else if the number is zero, print zero.
  - d. Finally compare the first and last element of the array and display if they equal, greater or less
3. Create a program to print a multiplication table of a number.

**Hint =>**

- a. Get an integer input and store it in the number variable. Also, define a integer array to store the results of multiplication from 1 to 1
  - b. Run a loop from 1 to 10 and store the results in the multiplication table array
  - c. Finally, display the result from the array in the format number \* i = \_\_\_\_
4. Write a program to store multiple values in an array up to a maximum of 10 or until the user enters a 0 or a negative number. Show all the numbers as well as the sum of all numbers

**Hint =>**

- a. Create a variable to store an array of 10 elements of type double as well as a variable to store the total of type double initializes to 0.0. Also, the index variable is initialized to 0 for the array
- b. Use infinite while loop as in **while (true)**
- c. Take the user entry and check if the user entered 0 or a negative number to break the loop
- d. Also, **break** from the loop if the index has a value of 10 as the array size is limited to 10.
- e. If the user entered a number other than 0 or a negative number inside the while loop then assign the number to the array element and increment the index value
- f. Take another **for** loop to get the values of each element and add it to the total
- g. Finally display the total value

5. Create a program to find the multiplication table of a number entered by the user from 6 to 9 and display the result

**Hint =>**

- a. Take integer input and store it in the variable number as well as define an integer array to store the multiplication result in the variable multiplicationResult
  - b. Using a for loop, find the multiplication table of numbers from 6 to 9 and save the result in the array
  - c. Finally, display the result from the array in the format number \* i = \_\_\_\_
6. Create a program to find the mean height of players present in a football team.

**Hint =>**

- a. The formula to calculate the mean is:  $\text{mean} = \text{sum of all elements} / \text{number of elements}$
  - b. Create a double array named heights of size 11 and get input values from the user.
  - c. Find the sum of all the elements present in the array.
  - d. Divide the sum by 11 to find the mean height and print the mean height of the football team
7. Create a program to save odd and even numbers into odd and even arrays between 1 to the number entered by the user. Finally, print the odd and even numbers array

**Hint =>**

- a. Get an integer input from the user, assign it to a variable **number**, and check for Natural Number. If not a natural number then print an error and exit the program
  - b. Create an integer array for even and odd numbers with `size = number / 2 + 1`
  - c. Create index variables for odd and even numbers and initialize them to zero
  - d. Using a for loop, iterate from 1 to the number, and in each iteration of the loop, save the odd or even number into the corresponding array
  - e. Finally, print the odd and even numbers array using the odd and even index
8. Create a program to find the factors of a number taken as user input, store the factors in an array, and display the factors

**Hint =>**

- a. Take the input for a number
- b. Find the factors of the number and save them in an array. For this create integer variable maxFactor and initialize to 10, factors array of size maxFactor and index variable to reflect the index of the array.
- c. To find factors loop through the numbers from 1 to the number, find the factors, and add them to the array element by incrementing the index. If the index is equal to maxIndex, then need factors array to store more elements
- d. To store more elements, reset the maxIndex to twice its size, use the temp array to store the elements from the factors array, and eventually assign the factors array to the temp array
- e. Finally, Display the factors of the number

9. Working with Multi-Dimensional Arrays. Write a Java program to create a 2D Array and Copy the 2D Array into a single dimension array

**Hint =>**

- Take user input for rows and columns, create a 2D array (Matrix), and take the user input
  - Copy the elements of the matrix to a 1D array. For this create a 1D array of size rows\*columns as in `int[] array = new int[rows * columns];`
  - Define the index variable and Loop through the 2D array. Copy every element of the 2D array into the 1D array and increment the index
  - Note: For looping through the 2D array, you will need Nested for loop, Outer for loop for rows, and the inner for loops to access each element
10. Write a program FizzBuzz, take a number as user input and if it is a positive integer loop from 0 to the number and save the number, but for multiples of 3 save "Fizz" instead of the number, for multiples of 5 save "Buzz", and for multiples of both save "FizzBuzz". Finally, print the array results for each index position in the format Position 1 = 1, ..., Position 3 = Fizz,...

**Hint =>**

- Create a String Array to save the results and
- Finally, loop again to show the results of the array based on the index position