# 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);
        }
```

```java
        // 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
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 2 Practice Programs

1. Create a program to find the bonus of 10 employees based on their years of service and the total bonus amount the company Zara has to pay, along with the old and new salary.

   **Hint =>**

   a. Zara decides to give a bonus of 5% to employees whose year of service is more than 5 years or 2% if less than 5 years
   b. Define a double array to save salary and years of service for each of the 10 employees
   c. Also define a double array to save the new salary and the bonus amount as well as variables to save the total bonus, total old salary, and new salary
   d. Define a loop to take input from the user. If salary or year of service is an invalid number then ask the use to enter again. Note in this case you will have to decrement the index counter
   e. Define another loop to calculate the bonus of 10 employees based on their years of service. Save the bonus in the array, compute the new salary, and save in the array. Also, the total bonus and total old and new salary can be calculated in the loop
   f. Print the total bonus payout as well as the total old and new salary of all the employees

```java
import java.util.Scanner;
public class EmployeeBonus {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int numEmployees = 10;
        double[] salaries = new double[numEmployees];
        double[] yearsOfService = new double[numEmployees];
        double[] bonuses = new double[numEmployees];
        double[] newSalaries = new double[numEmployees];

        double totalOldSalary = 0, totalBonus = 0, totalNewSalary = 0;

        for (int i = 0; i < numEmployees; i++) {
            while (true) {
                System.out.print("Enter salary of employee " + (i + 1) + ": ");
                salaries[i] = sc.nextDouble();
                System.out.print("Enter years of service: ");
                yearsOfService[i] = sc.nextDouble();

                if (salaries[i] > 0 && yearsOfService[i] >= 0) {
                    break;
                }
```

```
                System.out.println("Invalid input! Please enter again.");
            }
        }
        for (int i = 0; i < numEmployees; i++) {
            bonuses[i] = (yearsOfService[i] > 5) ? salaries[i] * 0.05 :
salaries[i] * 0.02;
            newSalaries[i] = salaries[i] + bonuses[i];

            totalOldSalary += salaries[i];
            totalBonus += bonuses[i];
            totalNewSalary += newSalaries[i];
        }
        System.out.println("\nTotal Bonus Payout: " + totalBonus);
        System.out.println("Total Old Salary: " + totalOldSalary);
        System.out.println("Total New Salary: " + totalNewSalary);

        sc.close();
    }
}
```

2. Create a program to find the youngest friends among 3 Amar, Akbar, and Anthony based on their ages and the tallest among the friends based on their heights

   **Hint =>**

   a. Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
   b. Loop through the array and find the youngest of the 3 friends and the tallest of the 3 friends
   c. Finally display the youngest and tallest of the 3 friends

```
import java.util.Scanner;

public class YoungestAndTallest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] names = {"Amar", "Akbar", "Anthony"};
        int[] ages = new int[3];
        double[] heights = new double[3];
        for (int i = 0; i < 3; i++) {
            System.out.print("Enter age of " + names[i] + ": ");
```

```
        ages[i] = sc.nextInt();
        System.out.print("Enter height (in cm) of " + names[i] + ": ");
        heights[i] = sc.nextDouble();
    }
    int minAgeIndex = 0;
    for (int i = 1; i < 3; i++) {
        if (ages[i] < ages[minAgeIndex]) {
            minAgeIndex = i;
        }
    }
    int maxHeightIndex = 0;
    for (int i = 1; i < 3; i++) {
        if (heights[i] > heights[maxHeightIndex]) {
            maxHeightIndex = i;
        }
    }

    System.out.println("\nYoungest friend: " + names[minAgeIndex] + "
(Age: " + ages[minAgeIndex] + ")");
    System.out.println("Tallest friend: " + names[maxHeightIndex] + "
(Height: " + heights[maxHeightIndex] + " cm)");
    sc.close();
    }
}
```

3. Create a program to store the digits of the number in an array and find the largest and second largest element of the array.

   **Hint =>**
   a. Create a number variable and Take user input.
   b. Define an array to store the digits. Set the size of the array to maxDigit variable initially set to 10
   c. Create an integer variable index with the value 0 to reflect the array index.
   d. Use a loop to iterate until the number is not equal to 0.
   e. Remove the last digit from the number in each iteration and add it to the array.
   f. Increment the index by 1 in each iteration and if the index count equals maxDigit then break out of the loop and the remaining digits are not added to the array
   g. Define variable to store largest and second largest digit and initialize it to zero
   h. Loop through the array and use conditional statements to find the largest and second largest number in the array

i. Finally display the largest and second-largest number

```java
import java.util.Scanner;
public class LargestSecondLargest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = sc.nextInt();

        int maxDigit = 10;
        int[] digits = new int[maxDigit];
        int index = 0;
        while (number != 0) {
            digits[index++] = number % 10;
            number /= 10;
            if (index == maxDigit) break;
        }
        int largest = -1, secondLargest = -1;
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                secondLargest = digits[i];
            }
        }

        System.out.println("Largest digit: " + largest);
        System.out.println("Second Largest digit: " + secondLargest);
        sc.close();
    }
}
```

4. Rework the program 2, especially the **Hint** f where if index equals maxDigit, we break from the loop. Here we want to modify to Increase the size of the array i,e maxDigit by 10 if the index is equal to maxDigit. This is done to consider all digits to find the largest and second-largest number

**Hint =>**

a. In Hint f inside the loop if the index is equal to maxDigit, increase maxDigit and make digits array to store more elements.
b. To do this, we need to create a new temp array of size maxDigit, copy from the current digits array the digits into the temp array, and assign the current digits array to the temp array
c. Now the digits array will be able to store all digits of the number in the array and then find the largest and second largest number

```java
import java.util.Scanner;

public class DynamicLargestSecondLargest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = sc.nextInt();

        int maxDigit = 10;
        int[] digits = new int[maxDigit];
        int index = 0;

        while (number != 0) {
            if (index == maxDigit) {
                maxDigit += 10;
                int[] temp = new int[maxDigit];
                System.arraycopy(digits, 0, temp, 0, digits.length);
                digits = temp;
            }
            digits[index++] = number % 10;
            number /= 10;
        }

        int largest = -1, secondLargest = -1;
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                secondLargest = digits[i];
            }
        }
    }
```

```
        System.out.println("Largest digit: " + largest);
        System.out.println("Second Largest digit: " + secondLargest);
        sc.close();
    }
}
```

5. Create a program to take a number as input and reverse the number. To do this, store the digits of the number in an array and display the array in reverse order

**Hint =>**

a. Take user input for a number.
b. Find the count of digits in the number.
c. Find the digits in the number and save them in an array
d. Create an array to store the elements of the digits array in reverse order
e. Finally, display the elements of the array in reverse order

```java
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = sc.nextInt();

        int count = (int) Math.log10(number) + 1;
        int[] digits = new int[count];

        for (int i = 0; i < count; i++) {
            digits[i] = number % 10;
            number /= 10;
        }

        System.out.print("Reversed number: ");
        for (int i = 0; i < count; i++) {
            System.out.print(digits[i]);
        }
        sc.close();
    }
}
```

6. An organization took up an exercise to find the Body Mass Index (BMI) of all the persons in the team. For this create a program to find the BMI and display the height, weight, BMI and status of each individual

**Hint =>**
a. Take input for a number of persons
b. Create arrays to store the weight, height, BMI, and weight status of the persons
c. Take input for the weight and height of the persons
d. Calculate the BMI of all the persons and store them in an array and also find the weight status of the persons
e. Display the height, weight, BMI, and weight status of each person
f. Use the table to determine the weight status of the person

| BMI | Status |
|---|---|
| ≤ 18.4 | Underweight |
| 18.5 - 24.9 | Normal |
| 25.0 - 39.9 | Overweight |
| ≥ 40.0 | Obese |

```java
import java.util.Scanner;

public class BMI_Calculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of persons: ");
        int n = sc.nextInt();

        double[] heights = new double[n];
        double[] weights = new double[n];
        double[] bmi = new double[n];
        String[] status = new String[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter height (in meters) for person " + (i +
1) + ": ");
            heights[i] = sc.nextDouble();
            System.out.print("Enter weight (in kg) for person " + (i + 1) +
": ");
```

```
        weights[i] = sc.nextDouble();

        bmi[i] = weights[i] / (heights[i] * heights[i]);

        if (bmi[i] <= 18.4) status[i] = "Underweight";
        else if (bmi[i] <= 24.9) status[i] = "Normal";
        else if (bmi[i] <= 39.9) status[i] = "Overweight";
        else status[i] = "Obese";
    }

    System.out.println("\nHeight(m)  Weight(kg)  BMI   Status");
    for (int i = 0; i < n; i++) {
        System.out.printf("%.2f      %.2f        %.2f   %s\n",
heights[i], weights[i], bmi[i], status[i]);
    }
    sc.close();
    }
}
```

7. Rewrite the above program using multi-dimensional array to store height, weight, and BMI in 2D array for all the persons

   **Hint =>**
   a. Take input for a number of persons
   b. Create a multi-dimensional array to store weight, height and BMI. Also create an to store the weight status of the persons

   ```
   double[][] personData = new double[number][3];
   String[] weightStatus = new String[number];
   ```

   c. Take input for weight and height of the persons and for negative values, ask the user to enter positive values
   d. Calculate BMI of all the persons and store them in the personData array and also find the weight status and put them in the weightStatus array
   e. Display the height, weight, BMI and status of each person

```
import java.util.Scanner;

public class BMI_2DArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of persons: ");
```

```java
        int n = sc.nextInt();

        double[][] personData = new double[n][3];
        String[] weightStatus = new String[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter height (in meters) for person " + (i +
1) + ": ");
            personData[i][0] = sc.nextDouble();
            System.out.print("Enter weight (in kg) for person " + (i + 1) +
": ");
            personData[i][1] = sc.nextDouble();

            personData[i][2] = personData[i][1] / (personData[i][0] *
personData[i][0]);

            weightStatus[i] = (personData[i][2] <= 18.4) ? "Underweight" :
                              (personData[i][2] <= 24.9) ? "Normal" :
                              (personData[i][2] <= 39.9) ? "Overweight" :
"Obese";
        }

        System.out.println("\nHeight(m)  Weight(kg)  BMI   Status");
        for (int i = 0; i < n; i++) {
            System.out.printf("%.2f       %.2f        %.2f   %s\n",
personData[i][0], personData[i][1], personData[i][2], weightStatus[i]);
        }
        sc.close();
    }
}
```

8. Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as per the following guidelines

| Grade | Remarks | Marks |
|-------|---------|-------|
| A | (Level 4, above agency-normalized standards) | 80% and above |
| B | (Level 3, at agency-normalized standards) | 70-79% |
| C | (Level 2, below, but approaching agency-normalized standards) | 60-69% |
| D | (Level 1, well below agency-normalized standards) | 50-59% |
| E | (Level 1- , too below agency-normalized standards) | 40-49% |
| R | (Remedial standards) | 39% and below |

**Hint =>**
a. Take input for the number of students
b. Create arrays to store marks, percentages, and grades of the students
c. Take input for marks of students in physics, chemistry, and maths. If the marks are negative, ask the user to enter positive values and decrement the index
d. Calculate the percentage and grade of the students based on the percentage
e. Display the marks, percentages, and grades of each student

```java
import java.util.Scanner;

public class StudentGradesArray {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int numStudents = scan.nextInt();


        int[] physics = new int[numStudents];
        int[] chemistry = new int[numStudents];
        int[] maths = new int[numStudents];
        double[] percentages = new double[numStudents];
        char[] grades = new char[numStudents];


        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter marks for Student " + (i + 1) +
":");

            physics[i] = getValidMarks(scan, "Physics");
            chemistry[i] = getValidMarks(scan, "Chemistry");
```

```java
            maths[i] = getValidMarks(scan, "Maths");
            percentages[i] = (physics[i] + chemistry[i] + maths[i]) / 3.0;

            grades[i] = calculateGrade(percentages[i]);
        }


        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            System.out.println("Physics: " + physics[i]);
            System.out.println("Chemistry: " + chemistry[i]);
            System.out.println("Maths: " + maths[i]);
            System.out.println("Percentage: " + percentages[i] + "%");
            System.out.println("Grade: " + grades[i]);
        }

        scan.close();
    }


    public static int getValidMarks(Scanner scan, String subject) {
        int marks;
        while (true) {
            System.out.print("Enter marks for " + subject + " (0-100): ");
            marks = scan.nextInt();
            if (marks >= 0 && marks <= 100) {
                break;
            } else {
                System.out.println("Invalid marks! Please enter a value
between 0 and 100.");
            }
        }
        return marks;
    }


    public static char calculateGrade(double percentage) {
        if (percentage >= 80) return 'A';
        else if (percentage >= 70) return 'B';
        else if (percentage >= 60) return 'C';
        else if (percentage >= 50) return 'D';
```

```
        else if (percentage >= 40) return 'E';
        else return 'R';
    }
}
```

9. Rewrite the above program to store the marks of the students in physics, chemistry, and maths in a 2D array and then compute the percentage and grade

   **Hint =>**
   a. All the steps are the same as the problem 8 except the marks are stored in a 2D array
   b. Use the 2D array to calculate the percentages, and grades of the students

```java
import java.util.Scanner;

public class StudentGrades2DArray {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int numStudents = scan.nextInt();


        int[][] marks = new int[numStudents][3];
        double[] percentages = new double[numStudents];
        char[] grades = new char[numStudents];
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter marks for Student " + (i + 1) +
":");

            marks[i][0] = getValidMarks(scan, "Physics");
            marks[i][1] = getValidMarks(scan, "Chemistry");
            marks[i][2] = getValidMarks(scan, "Maths");


            percentages[i] = (marks[i][0] + marks[i][1] + marks[i][2]) /
3.0;
```

```
            grades[i] = calculateGrade(percentages[i]);
        }
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            System.out.println("Physics: " + marks[i][0]);
            System.out.println("Chemistry: " + marks[i][1]);
            System.out.println("Maths: " + marks[i][2]);
            System.out.println("Percentage: " + percentages[i] + "%");
            System.out.println("Grade: " + grades[i]);
        }

        scan.close();
    }
    public static int getValidMarks(Scanner scan, String subject) {
        int marks;
        while (true) {
            System.out.print("Enter marks for " + subject + " (0-100): ");
            marks = scan.nextInt();
            if (marks >= 0 && marks <= 100) {
                break;
            } else {
                System.out.println("Invalid marks! Please enter a value
between 0 and 100.");
            }
        }
        return marks;
    }
    public static char calculateGrade(double percentage) {
        if (percentage >= 80) return 'A';
        else if (percentage >= 70) return 'B';
        else if (percentage >= 60) return 'C';
        else if (percentage >= 50) return 'D';
        else if (percentage >= 40) return 'E';
        else return 'R';
    }
}
```

10. Create a program to take a number as input find the frequency of each digit in the number using an array and display the frequency of each digit

**Hint =>**

a. Take the input for a number
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 frequency of each digit in the number. For this define a frequency array of size 10, Loop through the digits array, and increase the frequency of each digit
e. Display the frequency of each digit in the number

```java
import java.util.Scanner;

public class DigitFrequency {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter a number: ");
        long number = scan.nextLong();
        int[] frequency = new int[10];
        while (number > 0) {
            int digit = (int) (number % 10);
            frequency[digit]++;
            number /= 10;
        }

        System.out.println("Digit Frequency:");
        for (int i = 0; i < 10; i++) {
            if (frequency[i] > 0) {
                System.out.println("Digit " + i + ": " + frequency[i]);
            }
        }

        scan.close();
    }
}
```