Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Proper naming conventions for all variables

```java
String name = "Eric";
double height = input.nextDouble();
double totalDistance = distanceFromToVia + distanceViaToFinalCity;
```

3. Proper Program Name and Class Name
4. Follow proper indentation
5. Give comments for every step or logical block like a variable declaration or conditional and loop blocks

1. **Sample Program 1 -** Create a program to check if 3 values are internal angles of a triangle.

   IMP => Follow Good Programming Practice demonstrated below in all Practice Programs

   Hint =>

   a. Get integer input for 3 variables named x, y, and z.
   b. Find the sum of x, y, and z.
   c. If the sum is equal to 180, print "The given angles are internal angles of a triangle" else print They are not

```java
// Creating Class with name TriangleChecker indicating the purpose is to
// check if the internal angles add to 180
import java.util.Scanner;

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

        // Get 3 input values for angles
        int x = input.nextInt();
        int y = input.nextInt();
        int z = input.nextInt();

        // Find the sum of all angles
        int sumOfAngles = x + y + z;

        // Check if sum is equal to 180 and print either true or false
        System.out.println("The given angles " +x+ ", " +y+ ", " + z +
                           " add to " + sumOfAngles);
```

```java
        if (sumOfAngles == 180) {
            System.out.println("The given angles are internal angles of a " +
                               "Triangle");
        } else {
            System.out.println("The given angles are not internal angles " +
                               "of a Triangle");
        }

        // Closing the Scanner Stream
        input.close();
    }
}
```

2. **Sample Program 2 -** Create a program to find the sum of all the digits of a number given by a user.

   Hint =>

   a. Get an integer input for the number variable.
   b. Create an integer variable sum with an initial value of 0.
   c. Create a while loop to access each digit of the number.
   d. Inside the loop, add each digit of the number to the sum.
   e. Finally, print the sum outside the loop

```java
Java
// Create SunOfDigit Class to compute the sum of all digits of a number
import java.util.Scanner;

class SumOfDigits {

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

        // Get input value for number
        int origNumber = input.nextInt();

        // Define variable number and sum initialized to zero
        int number = origNumber;
        int sum = 0;
```

```java
        // Run while loop to access each digit of number
        while (number != 0) {
            // Use number % 10 to find each digit of number from last
            int digit = number % 10;

            // add each digit to sum
            sum += digit;

            // Remove last digit from number essentially get the quotient
            number = number / 10;
        }

        // Print the sum and close the Scanner Stream
        System.out.println("The sum of digit of number:" +origNumber+ " = " +
                        sum);
        input.close();
    }
}
```

Level 2 Practice Programs

1. Write a LeapYear program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year.

    Hint =>

    a. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar. So ensure to check for the same.
    b. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.
    c. Write code having multiple **if else** statements based on conditions provided above and a second part having only one if statement and multiple logical

2. Rewrite program 1 to determine Leap Year with single if condition using logical and **&&** and or **||** operators

3. Write a program to input marks and 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. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks

4. Write a Program to check if the given number is a prime number or not

    Hint =>
    a. A number that can be divided exactly only by itself and 1 are Prime Numbers,
    b. Prime Numbers checks are done for numbers greater than 1
    c. Loop through all the numbers from 2 to the user input number and check if the reminder is zero. If the reminder is zero break out from the loop as the number is divisible by some other number and is not a prime number.
    d. Use isPrime boolean variable to store the result

5. Write a program FizzBuzz, take a number as user input, and if it is a positive integer loop from 0 to the number and print the number, but for multiples of 3 print "Fizz" instead of the number, for multiples of 5 print "Buzz", and for multiples of both print "FizzBuzz".

   Hint =>
   a. Write the program and use *for* loop

6. Rewrite the program 5 FizzBuzz using while loop

7. Create a program to find the BMI of a person

   Hint =>
   a. Take user input in double for the weight (in kg) of the person and height (in cm) for the person and store it in the corresponding variable.
   b. Use the formula BMI = weight / (height * height). Note unit is kg/m^2. For this convert cm to meter
   c. 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

8. 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 the age and height of the 3 friends and store it in a variable
   b. Find the smallest of the 3 ages to find the youngest friend and display it
   c. Find the largest of the 3 heights to find the tallest friend and display it

9. Create a program to print the greatest factor of a number beside itself using a loop.

   Hint =>
   a. Get an integer input and assign it to the number variable. As well as define a greatestFactor variable and assign it to 1
   b. Create a *for* loop that runs from last but one till 1 as in i = number - 1 to i = 1.
   c. Inside the loop, check if the number is perfectly divisible by i then assign i to greatestFactor variable and break the loop.
   d. Display the greatestFactor variable outside the loop

10. Create a program to find the power of a number.

Hint =>

a. Get integer input for two variables named number and power.
b. Create a result variable with an initial value of 1.
c. Run a for loop from i = 1 to i <= power.
d. In each iteration of the loop, multiply the result with the number and assign the value to the result.
e. Finally, print the result

11. Create a program to find the factors of a number taken as user input.

Hint =>

a. Get the input value for a variable named number.
b. Run a *for* loop from i = 1 to i < number.
c. In each iteration of the loop, check if number is perfectly divisible by i.
d. If true, print the value of i.

12. Create a program to find all the multiple of a number taken as user input below 100.

Hint =>

a. Get input value for a variable named number.
b. Run a *for* loop backward: from i = 100 to i = 1.
c. Inside the loop, check if *i* perfectly divide the number.
d. If true, print the number and *continue* the loop.