

# Best Programming Practice

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

```
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);
```

```

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 =>**

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

Java

```

// Create SumOfDigit 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;
    }
}

```

```
// 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. Create a program to print odd and even numbers between 1 to the number entered by the user.

**Hint =>**

- a. Get an integer input from the user, assign to a variable number and check for Natural Number
  - b. Using a for loop, iterate from 1 to the number
  - c. In each iteration of the loop, print the number is odd or even number
2. Create a program to find the bonuses of employees based on their years of service.

**Hint =>**

- a. Zara decided to give a bonus of 5% to employees whose year of service is more than 5 years.
  - b. Take salary and year of service in the year as input.
  - c. Print the bonus amount.
3. Create a program to find the multiplication table of a number entered by the user from 6 to 9.

**Hint =>**

- a. Take integer input and store it in the variable number
  - b. Using a for loop, find the multiplication table of number from 6 to 9 and print it in the format number \* i = \_\_\_\_
4. Write a program FizzBuzz, take a number as user input, and check for a positive integer. If positive integer, loop 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. Take the user input number, check for a positive integer, and use **for** loop to display
5. Rewrite the program 4 FizzBuzz using the while loop
  6. 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
7. 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 and check if it is a positive integer.
- b. Run a **for** loop from i = 1 to i < number. In each iteration of the loop, check if the number is perfectly divisible by i. If true, print the value of i.

8. Rewrite the above program 7 to find the factors of a number using the **while** loop

**Hint =>**

- Get the input value for a variable named number and check if it is a positive integer.
- Create a counter variable and run the **while** loop till the counter is less than the user input number. In each iteration of the loop, check if the number is perfectly divisible by the counter. If true, print the value of the counter.

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

**Hint =>**

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

10. Rewrite the above program to print the greatest factor of a number beside itself using a **while** loop.

**Hint =>**

- Get an integer input and assign it to the number variable. As well as define a greatestFactor variable and assign it to 1
- Create a variable counter and assign **counter = number - 1**; Use the **while** loop till the counter is equal to 1.
- Inside the loop, check if the number is perfectly divisible by the counter then assign the counter to greatestFactor variable and break the loop.
- Display the greatestFactor variable outside the loop

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

**Hint =>**

- Get the input value for a variable named number. Check the number is a positive integer and less than 100.
- Run a **for** loop backward: from  $i = 100$  to  $i = 1$ .
- Inside the loop, check if  $i$  perfectly divide the number. If true, print the number and **continue** the loop.

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

**Hint =>**

- Get integer input for two variables - number and power and check for positive integer
- Create a result variable with an initial value of 1.
- Run a for loop from  $i = 1$  to  $i \leq \text{power}$ . In each iteration of the loop, multiply the result by the number and assign the value to the result. Finally, print the result

13. Rewrite the program to find all the multiples of a number below 100 using **while** loop.

**Hint =>**

- a. Get the input value for a variable named number. Check the number is a positive integer and less than 100.
- b. Create a counter variable and assign **counter = number - 1**; Use a **while** till the counter is > 1
- c. Inside the loop, check if the counter perfectly divides the number. If true, print the number and **continue** the loop.

14. Rewrite the above program to find the power of a number using a **while** loop.

**Hint =>**

- a. Get integer input for two variables named number and power.
- b. Create a result variable with an initial value of 1.
- d. Create a temp variable counter and initialize to zero. Use the **while** loop till **\_\_\*\*counter == power\*\*\_\_**.
- c. In each iteration of the loop, multiply the result by the number and assign the value to the result. Also, increment the counter.
- d. Finally, print the result

