CodInClub

Powered By BridgeLabz

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

```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 =>

   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
// 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;
```

```
        // 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 3 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 the isPrime boolean variable to store the result

5. Create a program to check if a number is armstrong or not. Use the hints to show the steps clearly in the code

   Hint =>

   a. Armstrong Number is a number whose Sum of cubes of each digit results in the original number as in for e.g. 153 = 1^3 + 5^3 + 3^3
   b. Get an integer input and store it in the number variable and define sum variable, initialize it to zero and originalNumber variable and assign it to input number variable
   c. Use the ***while*** loop till the originalNumber is not equal to zero
   d. In the ***while*** loop find each digit which is the reminder of the modulus operation ***number % 10***. Find the cube of the number and add it to the ***sum*** variable
   e. Again in while loop find the quotient of the number using the division operation ***number/10*** and assign it to the original number. This removes the last digit of the original number.
   f. Finally check if the number and the sum are the same, if same its an Armstrong number else not. So display accordingly

6. Create a program to count the number of digits in an integer.

   Hint =>

   a. Get an integer input for the number variable.
   b. Create an integer variable count with value 0.
   c. Use a loop to iterate until number is not equal to 0.
   d. Remove the last digit from number in each iteration
   e. Increase count by 1 in each iteration.
   f. Finally display the count to show the number of digits

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 check if a number taken from the user is a Harshad Number.

   Hint =>

   a. A Harshad number is an integer which is divisible by the sum of its digits.

      For example, 21 which is perfectly divided by 3 (sum of digits: 2 + 1).

   b. Get an integer input for the number variable.
   c. Create an integer variable sum with initial value 0.
   d. Create a while loop to access each digit of the number.
   e. Inside the loop, add each digit of the number to sum.
   f. Check if the number is perfectly divisible by the sum.
   g. If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

9. Create a program to check if a number is an Abundant Number.

   Hint =>

   a. An abundant number is an integer in which the sum of all the divisors of the number is greater than the number itself. For example,

      Divisor of 12: 1, 2, 3, 4, 6

      Sum of divisor: 1 + 2 + 3 + 4 + 6 = 16 > 12

   b. Get an integer input for the number variable.
   c. Create an integer variable sum with initial value 0.
   d. Run a for loop from i = 1 to i < number.
   e. Inside the loop, check if number is divisible by i.
   f. If true, add i to sum.
   g. Outside the loop Check if sum is greater than number.
   h. If the sum is greater than the number, print Abundant Number. Otherwise, print Not an Abundant Number.

10. Write a program to create a calculator using ***switch...case***.

    Hint =>

    a. Create two double variables named first and second and a String variable named op.
    b. Get input values for all variables.
    c. The input for the operator can only be one of the four values: "+", "-", "*" or "/".
    d. Run a for loop from i = 1 to i < number.
    e. Based on the input value of the op, perform specific operations using the ***switch...case*** statement and print the result.
    f. If op is +, perform addition between first and second; if it is -, perform subtraction and so on.
    g. If op is neither of those 4 values, print Invalid Operator.

11. Write a program ***DayOfWeek*** that takes a date as input and prints the day of the week that the date falls on. Your program should take three command-line arguments: m (month), d (day), and y (year). For m use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where / denotes integer division):

$$y_0 = y - (14 - m) / 12$$

$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 \times ((14 - m) / 12) - 2$$

$$d_0 = (d + x + 31m_0 / 12) \bmod 7$$