# Best Programming Practice

1. Use variables for all values, including inputs, fixed values, and results.
2. Proper naming conventions for all variables
3. Avoid hardcoding values.
4. Use meaningful variable names.
5. Properly name programs and classes.

String name = "Eric";

double height = Convert.ToDouble(Console.ReadLine());

double totalDistance = distanceFromToVia + distanceViaToFinalCity;

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

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

// Creating Class with name TriangleChecker indicating the purpose is to

// check if the internal angles add to 180

using System;

class TriangleChecker

{

static void Main(string[] args)

{

// Prompt the user for input

Console.WriteLine("Enter three angles of a triangle:");

// Get 3 input values for angles

int x = int.Parse(Console.ReadLine());

int y = int.Parse(Console.ReadLine());

int z = int.Parse(Console.ReadLine());

// Find the sum of all angles

int sumOfAngles = x + y + z;

// Check if sum is equal to 180 and print the result

Console.WriteLine($"The given angles {x}, {y}, {z} add to {sumOfAngles}");

if (sumOfAngles == 180)

{

Console.WriteLine("The given angles are internal angles of a Triangle.");

}

else

{

Console.WriteLine("The given angles are not internal angles of a Triangle.");

}

}

}



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

**Hint =>**

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

// Create SumOfDigits Class to compute the sum of all digits of a number

using System;

class SumOfDigits

{

static void Main(string[] args)

{

// Prompt the user for input

Console.WriteLine("Enter a number to calculate the sum of its digits:");

// Get input value for the number

int origNumber = int.Parse(Console.ReadLine());

// Define variable number and sum, initialized to zero

int number = origNumber;

int sum = 0;

// Run while loop to access each digit of the number

while (number != 0)

{

// Use number % 10 to find each digit of the number from the last

int digit = number % 10;

// Add each digit to sum

sum += digit;

// Remove the last digit from number (essentially get the quotient)

number = number / 10;

}

// Print the sum

Console.WriteLine($"The sum of digits of the number {origNumber} is {sum}");

}

}



# Level 3 Practice Programs

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // Prompt the user for input  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());    // creating variables to Store original number and sum  int originalNumber = number;   int sum = 0;    // Loop through each digit of the number  while (originalNumber != 0) {  int remainder = originalNumber % 10;   sum += (int)Math.Pow(remainder, 3);   originalNumber /= 10;   }   // Checking if the sum equals number  if (sum == number){  Console.WriteLine("{0} is an Armstrong number." , number);  }  else {  Console.WriteLine("{0} is not an Armstrong number." , number);  }  } } |
| --- |

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

**Hint =>**

1. Get an integer input for the number variable.
2. Create an integer variable count with value 0.
3. Use a loop to iterate until number is not equal to 0.
4. Remove the last digit from number in each iteration
5. Increase count by 1 in each iteration.
6. Finally display the count to show the number of digits

| using System;  class Solution {  public static void Main() {  // Prompt the user for input  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());;    // taking absolute value of number  number = Math.Abs(number);    // Initialize count to 0  int count = 0;   // Count the digits  if (number == 0) {  count = 1;  }  else {   while (number != 0) {  // Remove the last digit  number /= 10;     count++;   }  }    // printing the result  Console.WriteLine("The number {0} has {1} digits" , number , digitCount);  } } |
| --- |

1. Create a program to check if a number taken from the user is a Harshad Number.

**Hint =>**

1. 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).

1. Get an integer input for the number variable.
2. Create an integer variable sum with initial value 0.
3. Create a while loop to access each digit of the number.
4. Inside the loop, add each digit of the number to sum.
5. Check if the number is perfectly divisible by the sum.
6. If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

| using System;  class Solution {  public static void Main() {  // Prompt the user for input  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());   // initializing variables   int sum = 0;  int originalNumber = number;   // Calculating the sum of the digits  while (number != 0) {  sum += number % 10;   number /= 10;   }   // Checking if the number is a Harshad number  if (originalNumber % sum == 0) {  Console.WriteLine("{0} is a Harshad Number" , originalNumber);  }  else {  Console.WriteLine("{0} is not a Harshad Number" , originalNumber);  }  } } |
| --- |

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

**Hint =>**

1. 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

1. Get an integer input for the number variable.
2. Create an integer variable sum with initial value 0.
3. Run a for loop from i = 1 to i < number.
4. Inside the loop, check if number is divisible by i.
5. If true, add i to sum.
6. Outside the loop Check if sum is greater than number.
7. If the sum is greater than the number, print Abundant Number. Otherwise, print Not an Abundant Number.

| using System;  class Solution {  public static void Main() {  // Prompt the user for input  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());    // initializing sum variable  int sum = 0;   // computing the sum of divisors of the number  for (int i = 1; i < number; i++) {  if (number % i == 0) {  sum += i;   }  }    // Checking if the number is an Abundant number  if (sum > number) {  Console.WriteLine("{0} is an Abundant Number" , number);  }  else {  Console.WriteLine("{0} is not an Abundant Number." , number);  }  } } |
| --- |

1. 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 × ((14 − *m*) / 12) − 2

*d*0 = (*d* + *x* + 31*m*0 / 12) mod 7

| using System;  class DayOfWeek {  public static void Main(string[] args) {  // Parsing input arguments  // Month  int m = Convert.ToInt32(args[0]);     // Day  int d = Convert.ToInt32(args[1]);     // Year  int y = Convert.ToInt32(args[2]);     // Validate the inputs  if (m < 1 || m > 12 || d < 1 || d > 31) {  Console.WriteLine("Invalid input");  }   // Computing the day of the week  int y0 = y - (14 - m) / 12;  int x = y0 + y0 / 4 - y0 / 100 + y0 / 400;  int m0 = m + 12 \* ((14 - m) / 12) - 2;  int d0 = (d + x + 31 \* m0 / 12) % 7;   // computing and printing the result  string day = d0 switch {  0 => "Sunday",  1 => "Monday",  2 => "Tuesday",  3 => "Wednesday",  4 => "Thursday",  5 => "Friday",  6 => "Saturday"  };  Console.WriteLine("{0} falls on the {1}-{2}-{3}" , day , d , m , y );  } } |
| --- |

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // Prompt the user for the first number  Console.Write("Enter the first number: ");  double first = Convert.ToDouble(Console.ReadLine());   // Prompt the user for the second number  Console.Write("Enter the second number: ");  double second = Convert.ToDouble(Console.ReadLine());   // Prompt the user for the operator  Console.Write("Enter an operator (+, -, \*, /): ");  string op = Console.ReadLine();   // Performing the operation to calculate and print result  switch (op) {  case "+":  Console.WriteLine("Result: {0} + {1} = {2}", first, second, first + second);  break;   case "-":  Console.WriteLine("Result: {0} - {1} = {2}", first, second, first - second);  break;   case "\*":  Console.WriteLine("Result: {0} \* {1} = {2}", first, second, first \* second);  break;   case "/":  if (second != 0) {  Console.WriteLine("Result: {0} / {1} = {2}", first, second, first / second);  }  else {  Console.WriteLine("Division by zero is not allowed");  }  break;   default:  Console.WriteLine("Invalid operator! (use: +, -, \*, /)");  break;  }  } } |
| --- |