# Best Programming Practices in C#

* All values as variables including Fixed, User Inputs, and Results
* Proper naming conventions for all variables
* Proper Program Name and Class Name
* Proper Method Name which indicates action taking inputs and providing result

**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.

1. Use Math.random() and get a 4-digit random integer number
2. Write a method to count digits in the number
3. Write a method to return an array of digits from a given number.
4. Write a method to Find the sum of the digits of the number in the array
5. Finally, display the sum of the digits of the number

using System;

class SumOfDigits

{

// Generate a 4-digit random number

public int Get4DigitRandomNumber()

{

Random random = new Random();

return random.Next(1000, 10000); // Generates a random number between 1000 and 9999

}

// Count the number of digits in a given number

public int CountDigits(int number)

{

int count = 0;

while (number > 0)

{

count++;

number /= 10;

}

return count;

}

// Extract digits from the number and store them in an array

public int[] GetDigits(int number, int count)

{

int[] digits = new int[count];

for (int i = count - 1; i >= 0; i--)

{

digits[i] = number % 10;

number /= 10;

}

return digits;

}

// Calculate the sum of the digits in the array

public int SumArray(int[] array)

{

int sum = 0;

foreach (int digit in array)

{

sum += digit;

}

return sum;

}

static void Main(string[] args)

{

// Instantiate the SumOfDigits class

SumOfDigits sumOfDigits = new SumOfDigits();

// Generate a 4-digit random number

int number = sumOfDigits.Get4DigitRandomNumber();

Console.WriteLine("The Random Number is: " + number);

// Count the number of digits

int count = sumOfDigits.CountDigits(number);

Console.WriteLine("The Count of Digits is: " + count);

// Extract digits into an array

int[] digits = sumOfDigits.GetDigits(number, count);

// Calculate the sum of the digits

int sum = sumOfDigits.SumArray(digits);

// Display the sum of the digits

Console.WriteLine("Sum of Digits: " + sum);

}

}



# Level 1 Practice Programs

1. Write a program to input the Principal, Rate, and Time values and calculate Simple Interest.

**Hint =>**

1. Simple Interest = Principal \* Rate \* Time / 100
2. Take user input for principal, rate, time
3. Write a method to calculate the simple interest given principle, rate and time as parameters
4. Output “The Simple Interest is \_\_\_ for Principal \_\_\_, Rate of Interest \_\_\_ and Time \_\_\_”

| using System;  class Solution {  // Method to calculate simple interest  public static double CalculateSimpleInterest(double principal, double rate, double time)  {  return (principal \* rate \* time) / 100;  }   public static void Main() {    // prompt user for principal, interest, time input  Console.Write("Enter Principal: ");  double principal = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter Rate of Interest: ");  double rate = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter Time (in years): ");  double time = Convert.ToDouble(Console.ReadLine());    // calculate and print simple Interest  double simpleInterest = CalculateSimpleInterest(principal, rate, time);  Console.WriteLine("The Simple Interest is {0} for Principal {1}, Rate of Interest {2}, and Time {3}", simpleInterest, principal, rate, time);  } } |
| --- |

1. Create a program to find the maximum number of handshakes among N number of students.

**Hint =>**

1. Get integer input for number of students
2. Use the combination = (n \* (n - 1)) / 2 formula to calculate the maximum number of possible handshakes.
3. Write a method to use the combination formulae to calculate the number of handshakes

| using System;  class Solution {  // Method to calculate maximum number of handshakes  public static int CalculateHandshakes(int n) {  return (n \* (n - 1)) / 2;  }   public static void Main() {  // prompt user for number of students input  Console.Write("Enter the number of students: ");  int n = Convert.ToInt32(Console.ReadLine());   int handshakes = CalculateHandshakes(n);  Console.WriteLine("Maximum number of handshakes among {0} students is {1}", n, handshakes);  } } |
| --- |

1. Create a program to find the maximum number of handshakes among N number of students.

**Hint =>**

1. Get integer input for numberOfStudents variable.
2. Use the combination = (n \* (n - 1)) / 2 formula to calculate the maximum number of possible handshakes.
3. Display the number of possible handshakes.

| using System;  class Solution {  // Method to calculate maximum number of handshakes  public static int CalculateHandshakes(int n) {  return (n \* (n - 1)) / 2;  }   public static void Main() {  // prompt user for number of students input  Console.Write("Enter the number of students: ");  int n = Convert.ToInt32(Console.ReadLine());   int handshakes = CalculateHandshakes(n);  Console.WriteLine("Maximum number of handshakes among {0} students is {1}", n, handshakes);  } } |
| --- |

1. An athlete runs in a triangular park with sides provided as input by the user in meters. If the athlete wants to complete a 5 km run, then how many rounds must the athlete complete

**Hint =>**

1. Take user input for 3 sides of a triangle
2. The perimeter of a triangle is the addition of all sides and rounds is distance/perimeter
3. Write a Method to compute the number of rounds user needs to do to complete 5km run

| using System;  class Solution {  // Method to calculate the number of rounds to cover a distance  public static int CalculateRounds(double side1, double side2, double side3, double distance) {  double perimeter = side1 + side2 + side3;  return (int)Math.Ceiling(distance / perimeter);  }   public static void Main() {    // prompt user for 3 sides input  Console.Write("Enter side1 (in meters): ");  double side1 = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter side2 (in meters): ");  double side2 = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter side3 (in meters): ");  double side3 = Convert.ToDouble(Console.ReadLine());   const double distance = 5000;  int rounds = CalculateRounds(side1, side2, side3, distance);    // print result  Console.WriteLine("The athlete needs to complete {0} rounds to finish 5 km.", rounds);  } } |
| --- |

1. Write a program to check whether a number is positive, negative, or zero.

**Hint =>** Get integer input from the user. Write a Method to return -1 for negative number, 1 for positive number and 0 if number is zero

| using System;  class Solution {  // Method to check if a number is positive, negative, or zero  public static int CheckNumber(int number) {  if (number > 0) return 1;  if (number < 0) return -1;  return 0;  }   public static void Main() {  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());   int result = CheckNumber(number);  string type = result == 1 ? "Positive" : result == -1 ? "Negative" : "Zero";  Console.WriteLine("The number is {0}.", type);  } } |
| --- |

1. Write a program SpringSeason that takes two int values month and day from the command line and prints “Its a Spring Season” otherwise prints “Not a Spring Season”.

**Hint =>** Spring Season is from March 20 to June 20. Write a Method to check for Spring season and return a boolean true or false

| using System;  class Solution {  // Method to determine if a date falls in spring season  public static bool IsSpringSeason(int month, int day) {  return ((month == 3 && day >= 20 && day <= 31) || (month == 4 && day >= 20 && day <= 30) || (month == 5 && day >= 20 && day <= 31) || (month == 6 && day <= 20)) ;  }   public static void Main() {  Console.Write("Enter month (1-12): ");  int month = Convert.ToInt32(Console.ReadLine());   Console.Write("Enter day (1-31): ");  int day = Convert.ToInt32(Console.ReadLine());   if (IsSpringSeason(month, day)) {  Console.WriteLine("It's a Spring Season.");  }  else {  Console.WriteLine("Not a Spring Season.");  }  } } |
| --- |

1. Write a program to find the sum of n natural numbers using loop

**Hint =>** Get integer input from the user. Write a Method to find the sum of n natural numbers using loop

| using System;  class Solution {  // Method to calculate the sum of the first N natural numbers  public static int CalculateSum(int n) {  int sum = 0;  for (int i = 1; i <= n; i++) {  sum += i;  }  return sum;  }   public static void Main() {  Console.Write("Enter a number: ");  int n = Convert.ToInt32(Console.ReadLine());   int sum = CalculateSum(n);  Console.WriteLine("The sum of the first {0} natural numbers is {1}.", n, sum);  } } |
| --- |

8.Write a program to find the smallest and the largest of the 3 numbers.

**Hint =>**

* Take user input for 3 numbers
* Write a single method to find the smallest and largest of the three numbers

***public static int[] FindSmallestAndLargest(int number1, int number2, int number3)***

| using System; using Math;  class Solution {  // Method to find smallest and largest numbers  public static int[] FindSmallestAndLargest(int num1, int num2, int num3) {  int smallest = Min(num1, Min(num2, num3));  int largest = Max(num1, Max(num2, num3));  return new int[] { smallest, largest };  }   public static void Main() {  Console.Write("Enter first number: ");  int num1 = Convert.ToInt32(Console.ReadLine());   Console.Write("Enter second number: ");  int num2 = Convert.ToInt32(Console.ReadLine());   Console.Write("Enter third number: ");  int num3 = Convert.ToInt32(Console.ReadLine());   int[] result = FindSmallestAndLargest(num1, num2, num3);  Console.WriteLine("Smallest: {0}, Largest: {1}.", result[0], result[1]);  } } |
| --- |

9.Write a program to take 2 numbers and print their quotient and reminder

**Hint =>**

* Take user input as integer
* Use division operator (/) for quotient and moduli operator (%) for reminder
* Write Method to find the reminder and the quotient of a number

***public static int[] FindRemainderAndQuotient(int number, int divisor)***

| using System; class Solution {  // Method to calculate quotient and remainder  public static int[] FindRemainderAndQuotient(int number, int divisor) {  int quotient = number / divisor;  int remainder = number % divisor;  return new int[] { quotient, remainder };  }   public static void Main() {  Console.Write("Enter number: ");  int number = Convert.ToInt32(Console.ReadLine());   Console.Write("Enter divisor: ");  int divisor = Convert.ToInt32(Console.ReadLine());   int[] result = FindRemainderAndQuotient(number, divisor);  Console.WriteLine("Quotient: {0}, Remainder: {1}.", result[0], result[1]);  } } |
| --- |

10.Create a program to divide N number of chocolates among M children. Print the number of chocolates each child will get and also the remaining chocolates

**Hint =>**

* Get an integer value from the user for the numberOfchocolates and numberOfChildren.
* Write the method to find the number of chocolates each child gets and number of remaining chocolates

***public static int[] FindRemainderAndQuotient(int number, int divisor)***

| using System;  class Solution {  // Method to calculate quotient and remainder  public static int[] FindRemainderAndQuotient(int number, int divisor) {  int quotient = number / divisor;  int remainder = number % divisor;  return new int[] { quotient, remainder };  }   public static void Main() {  Console.Write("Enter the number of chocolates: ");  int numberOfChocolates = Convert.ToInt32(Console.ReadLine());   Console.Write("Enter the number of children: ");  int numberOfChildren = Convert.ToInt32(Console.ReadLine());   // Check if divisor (number of children) is zero  if (numberOfChildren == 0) {  Console.WriteLine("Number of children cannot be zero.");  }  else {  // Calculate and display the results  int[] result = FindRemainderAndQuotient(numberOfChocolates, numberOfChildren);  Console.WriteLine("Each child gets {0} chocolates, and {1} chocolates remain.", result[0], result[1]);  }  } } |
| --- |

11.Write a program calculate the wind chill temperature given the temperature and wind speed

**Hint =>**

1. Write a method to calculate the wind chill temperature using the formula

windChill = 35.74 + 0.6215 \*temp + (0.4275\*temp - 35.75) \* windSpeed0.16

***public double CalculateWindChill(double temperature, double windSpeed)***

| using System;  class Solution {  // Method to calculate wind chill temperature  public static double CalculateWindChill(double temperature, double windSpeed) {  return 35.74 + (0.6215 \* temperature) + ((0.4275 \* temperature - 35.75) \* Math.Pow(windSpeed, 0.16));  }   public static void Main() {  Console.Write("Enter the temperature in Fahrenheit: ");  double temperature = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter the wind speed in mph: ");  double windSpeed = Convert.ToDouble(Console.ReadLine());   // Validate wind speed  if (windSpeed < 0) {  Console.WriteLine("Wind speed cannot be negative.");  }  else {  // Calculate and display wind chill temperature  double windChill = CalculateWindChill(temperature, windSpeed);  Console.WriteLine("The wind chill temperature is {0:F2}°F.", windChill);  }  } } |
| --- |

12.Write a program to calculate various trigonometric functions using Math class given an angle in degrees

**Hint =>**

1. Method to calculate various trigonometric functions, Firstly convert to radians and then use Math function to find sine, cosine and tangent.

***public double[] calculateTrigonometricFunctions(double angle)***

| using System;  class Solution {  // Method to calculate trigonometric functions  public static double[] CalculateTrigonometricFunctions(double angle) {  // Convert angle from degrees to radians  double radians = Math.PI \* angle / 180.0;  double sine = Math.Sin(radians);  double cosine = Math.Cos(radians);  double tangent = Math.Tan(radians);   return new double[] { sine, cosine, tangent };  }    public static void Main() {  Console.Write("Enter an angle in degrees: ");  double angle = Convert.ToDouble(Console.ReadLine());   // Calculate and display trigonometric values  double[] results = CalculateTrigonometricFunctions(angle);   Console.WriteLine("Sine: {0}", results[0]);  Console.WriteLine("Cosine: {0}", results[1]);  Console.WriteLine("Tangent: {0}", results[2]);  } } |
| --- |