**Best Programming Practices**

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

* String name = "Eric";
* double height = Convert.ToDouble(Console.ReadLine());
* double totalDistance = distanceFromToVia + distanceViaToFinalCity;

1. Maintain proper indentation.

**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:**

* Get integer input for 3 variables named x, y, and z.
* Find the sum of x, y, and z.
* If the sum is equal to 180, print “The given angles are internal angles of a triangle”; otherwise, 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:**

* Get an integer input for the number variable.
* Create an integer variable sum with an initial value of 0.
* Use a while loop to access each digit of the number.
* Inside the loop, add each digit of the number to sum.
* 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 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 =>**

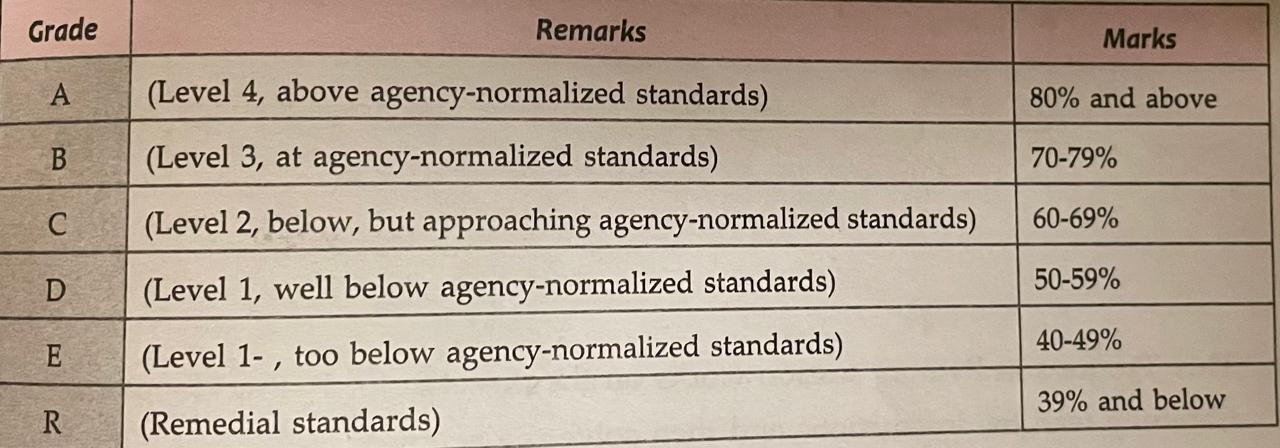
1. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar. So ensure to check for the same.
2. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.
3. Write code having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

| using System;  class Solution {  public static void CheckLeapYearUsingIfElse(int year) {  // using multiple if-else blocks to compute result  if (year % 4 == 0) {  if (year % 100 == 0) {  if (year % 400 == 0) {  Console.WriteLine("{0} is a Leap Year" , year);  } else {  Console.WriteLine("{0} is not a Leap Year" , year);  }  } else {  Console.WriteLine("{0} is a Leap Year" , year);  }  } else {  Console.WriteLine("{0} is not a Leap Year" , year);  }  }   public static void CheckLeapYearUsingLogicalOperators(int year) {  // using multiple LogicalOperators to compute result  if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {  Console.WriteLine("{0} is a Leap Year" , year);  } else {  Console.WriteLine("{0} is not a Leap Year" , year);  }  }  public static void Main() {   // prompt for user input  Console.Write("Enter a year (>= 1582): ");  int year = int.Parse(Console.ReadLine());    // checking year is greater than 1582 or not  if (year < 1582) {  Console.WriteLine("Year must be greater than or equal to 1582.");  } else {  // Using multiple if-else statements  Console.WriteLine("Using multiple if-else statements:");  CheckLeapYearUsingIfElse(year);    // Using a single if statement with logical operators  Console.WriteLine("Using a if statement and multiple logical :");  CheckLeapYearUsingLogicalOperators(year);  }   } } |
| --- |

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

| using System;  class Solution {  public static void Main() {  // prompt for user input  Console.Write("Enter a year (>= 1582): ");  int year = int.Parse(Console.ReadLine());   //computing and printing result  if (year < 1582) {  Console.WriteLine("Year must be greater than or equal to 1582.");  } else {  if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {  Console.WriteLine("{0} is a Leap Year" , year);  } else {  Console.WriteLine("{0} is not a Leap Year" , year);  }  }   } } |
| --- |

1. 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

****

**Hint =>**

1. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks

| using System;  class Solution {  public static void Main() {  // Prompt for user input for physics marks  Console.Write("Enter marks for Physics out of 100: ");  int physics = Convert.ToInt32(Console.ReadLine());   // Prompt for user input for Chemistry marks  Console.Write("Enter marks for Chemistry out of 100: ");  int chemistry = Convert.ToInt32(Console.ReadLine());   // Prompt for user input for Mathematics marks  Console.Write("Enter marks for Mathematics out of 100: ");  int maths = Convert.ToInt32(Console.ReadLine());   // Calculating total, percentage, and grade  int totalMarks = physics + chemistry + maths;  double percentage = totalMarks / 3.0;   // Computing grade  string grade = percentage switch {  >= 80 => "A",  >= 70 => "B",  >= 60 => "C",  >= 50 => "D",  >= 40 => "E",  \_ => "R"  };   // Computing remark  string remark = percentage switch {  >= 80 => "Level 4, above agency-normalized standards",  >= 70 => "Level 3, at agency-normalized standards",  >= 60 => "Level 2, below, but approaching agency-normalized standards",  >= 50 => "Level 1, well below agency-normalized standards",  >= 40 => "Level 1-, too below agency-normalized standards",  \_ => "Remedial standards"  };   // Printing results  Console.WriteLine("average marks: {0}", percentage);  Console.WriteLine("Grade: {0}", grade);  Console.WriteLine("Remark: {0}", remark);  } } |
| --- |

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

**Hint =>**

1. A number that can be divided exactly only by itself and 1 are Prime Numbers,
2. Prime Numbers checks are done for numbers greater than 1
3. 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.
4. Use isPrime boolean variable to store the result

| using System;  class Solution {  public static void Main() {  // prompt for user input  Console.Write("Enter a number greater than 1: ");  int number = Convert.ToInt32(Console.ReadLine());    // create boolean variable to check number is prime  bool isPrime = true;   // Loop through numbers from 2 to the square root of the number  for (int i = 2; i <= Math.Sqrt(number); i++) {  if (number % i == 0) {  isPrime = false;   // Exit loop if number is not prime  break;  }  }   // printing result  if (isPrime) {  Console.WriteLine("{0} is a Prime Number" , number);  } else {  Console.WriteLine("{0} is not a Prime Number" , number);  }  } } |
| --- |

1. 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 =>**

1. Write the program and use ***for*** loop

| using System;  class Solution {  public static void Main() {  // prompt for user input  Console.Write("Enter a positive number: ");  int number = Convert.ToInt32(Console.ReadLine());   // computing and printing result  for (int i = 0; i <= number; i++) {  if (i % 3 == 0 && i % 5 == 0) {  Console.WriteLine("FizzBuzz");  }  else if (i % 3 == 0) {  Console.WriteLine("Fizz");  }  else if (i % 5 == 0) {  Console.WriteLine("Buzz");  }  else{  Console.WriteLine(i);  }  }  } } |
| --- |

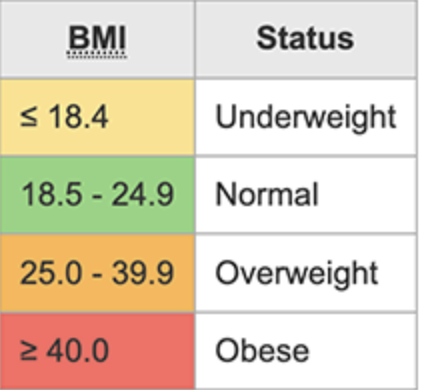
1. Rewrite the program 5 FizzBuzz using while loop

| using System;  class Solution {  public static void Main() {  // prompt for user input  Console.Write("Enter a positive number: ");  int number = Convert.ToInt32(Console.ReadLine());    // initializing flag variable  int i=0;  // computing and printing result  while (i <= number) {  if (i % 3 == 0 && i % 5 == 0) {  Console.WriteLine("FizzBuzz");  }  else if (i % 3 == 0) {  Console.WriteLine("Fizz");  }  else if (i % 5 == 0) {  Console.WriteLine("Buzz");  }  else{  Console.WriteLine(i);  }  i++;  }  } } |
| --- |

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

**Hint =>**

1. 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.
2. Use the formula BMI = weight / (height \* height). Note unit is kg/m^2. For this convert cm to meter
3. Use the table to determine the weight status of the person



| using System;  class Solution {  public static void Main() {  Console.Write("Enter weight in kilograms (kg): ");  double weight = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter height in centimeters (cm): ");  double heightInCm = Convert.ToDouble(Console.ReadLine());   // Converting height from cm to meters  double heightInMeters = heightInCm / 100;   // Calculating BMI  double bmi = weight / (heightInMeters \* heightInMeters);   // compute weight status based on BMI  string weightStatus = bmi switch {  <= 18.4 => "Underweight",  >= 18.5 and <= 24.9 => "Normal weight",  >= 25 and <= 39.9 => "Overweight",  \_ => "Obesity"  };   // printing results  Console.WriteLine("BMI: {0} kg/m^2 " , bmi);  Console.WriteLine("Weight Status: {0}" , weightStatus);  } } |
| --- |

1. 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 =>**

1. Take user input for the age and height of the 3 friends and store it in a variable
2. Find the smallest of the 3 ages to find the youngest friend and display it
3. Find the largest of the 3 heights to find the tallest friend and display it

| using System;  class Solution {  public static void Main() {  // prompt user for age and height for Amar input  Console.Write("Enter Amar's age: ");  int ageAmar = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Amar's height (in cm): ");  int heightAmar = Convert.ToInt32(Console.ReadLine());   // prompt user for age and height for Akbar input  Console.Write("Enter Akbar's age: ");  int ageAkbar = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Akbar's height (in cm): ");  int heightAkbar = Convert.ToInt32(Console.ReadLine());   // prompt user for age and height for Anthony input  Console.Write("Enter Anthony's age: ");  int ageAnthony = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Anthony's height (in cm): ");  int heightAnthony = Convert.ToInt32(Console.ReadLine());  // creating variables to store youngest and tallest  string youngest = "";  string tallest = "";    // checking youngest   if (ageAmar < ageAkbar && ageAmar < ageAnthony) {  youngest = "Amar";  }  else if (ageAkbar < ageAnthony) {  youngest = "Akbar";  }  else {  youngest = "Anthony";  }   // checking the tallest   if (heightAmar > heightAkbar && heightAmar > heightAnthony) {  tallest = "Amar";  }  else if (heightAkbar > heightAnthony) {  tallest = "Akbar";  }  else {  tallest = "Anthony";  }   // Printing results  Console.WriteLine("The youngest friend is {0} " , youngest);  Console.WriteLine("The tallest friend is {0} " , tallest);  } } |
| --- |

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // prompt user for number input  Console.Write("Enter a number: ");  int number = Convert.ToInt32(Console.ReadLine());   // Initialize the variable  int greatestFactor = 1;   // computing greatest Factor  for (int i = number - 1; i >= 1; i--) {  // Checking if the number is perfectly divisible by i  if (number % i == 0) {  greatestFactor = i;    // Exit the loop  break;  }  }   // Printing result  Console.WriteLine("The greatest factor of {0} is: {1}", number, greatestFactor);  } } |
| --- |

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // prompt for user number input  Console.Write("Enter the base number: ");  int number = Convert.ToInt32(Console.ReadLine());   // prompt for user power input  Console.Write("Enter the power: ");  int power = Convert.ToInt32(Console.ReadLine());   // Initialize result variable  int result = 1;   // Loop to calculate the power  for (int i = 1; i <= power; i++) {  // Multiply result by the i in each iteration  result \*= number;   }   // printing the result  Console.WriteLine("{0} raised to the power of {1} is: {2}", number, power, result);  } } |
| --- |

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // prompt for user input  Console.Write("Enter a number : ");  int number = Convert.ToInt32(Console.ReadLine());   // computing and printing the factors of number using for loop  Console.WriteLine("The factors of {0} are:", number);  for (int i = 1; i <= number; i++) {  if (number % i == 0) {  Console.WriteLine(i);   }  }  } } |
| --- |

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

**Hint =>**

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

| using System;  class Solution {  public static void Main() {  // prompt user for input  Console.Write("Enter a number : ");  int number = Convert.ToInt32(Console.ReadLine());   // computing and printing the result using for loop  for (int i = 100; i >= 1; i--) {  if (number % i == 0) {  Console.WriteLine(i);   }  }  } } |
| --- |