**Practical File**

****

Year - 2018-2021

**C#.NET**

**Submitted To: Submitted By:**

Mr. Chandrasekhar Patel Swasti Krishna

Lecturer BCA (5th Semester)

Department of Computer Science Department of Computer Science

**Department of Computer Science,**

**Dev Sanskriti Vishwavidyalaya**

**Gayatrikunj-Shantikunj, Haridwar, U.K. -249411,**

[**www.dsvv.ac.in**](http://www.dsvv.ac.in/)

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Task** | **Page No** |
| 1 | Write a program to print Armstrong Numbers | 3-4 |
| 2 | Write a program to print factorial of a number | 5 |
| 3 | Write a program to find the GCD of two numbers | 6 |
| 4 | Write a program to check if a number is prime number | 7-8 |
| 5 | Write a program to print the Fibonacci series | 9-10 |
| 6 | Write a program to print the half pyramid pattern | 11 |
| 7 | Write a program to print the half pyramid pattern with numbers | 12 |
| 8 | Write a program to print the half pyramid inverse pattern | 13 |
| 9 | Write a program to print the pyramid pattern | 14 |
| 10 | Write a program to print the inverse pyramid pattern | 15 |
| 11 | Write a program to print the diamond pattern | 16-17 |
| 12 | Write a program to print the Pascal’s triangle | 18-19 |
| 13 | Write a program to compare two string without using string library functions | 20-21 |
| 14 | Write a program to count a total number of alphabets, digits and special characters in a string | 22 |
| 15 | Write a program to copy one string to another string | 23 |
| 16 | Write a program to find maximum occurring character in a string | 24 |
| 17 | Write a program to check whether a given substring is present in the given string | 25-26 |
| 18 | Write a program for Encapsulation | 27 |
| 19 | Write a program for Abstraction | 28-29 |
| 20 | Write a program for single Inheritence | 30 |
| 21 | Write a program for Multilevel Inheritence | 31 |
| 22 | Write a program for Multiple Inheritence | 32 |
| 23 | Write a program for method overloading | 33 |
| 24 | Write a program for method overriding | 34 |
| 25 | Write a program for Interface | 35 |
| 26 | Write a program for Namespace | 36 |
| 27 | Write a program for exception handling through try and catch | 37 |
| 28 | Write a program for constructor | 38-39 |
| 29 | Write a program for Properties | 40 |
| 30 | Write a program for Threading | 41 |
| 31 | Write a program for Indexer | 42 |
| 32 | Write a program to access data from database using ADO.NET | 43 |

**………………………**

**Signature**

**1.** **Write a program to print Armstrong Numbers**

using System;

namespace myproject

{

public class Armstrong

{

public static void Main(string[] args)

{

Console.WriteLine("A program to check the given number is Armstrong Number or not");

int n, m, num, d;

double result = 0, number;

Console.WriteLine("Enter the number of iterations: ");

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

for (m = 0; m < n; m++)

{

Console.WriteLine("Enter the number of digits: ");

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

Console.WriteLine("Enter the number: ");

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

number = num;

for (int i = 0; i < d; i++)

{

int rem = num % 10; //split last digit from number

double power = Math.Pow(rem, d);

//Console.WriteLine(power);

result = result + power;

num = num / 10;

}

if (number == result)

{

Console.WriteLine(number + " Armstrong number.");

result = 0;

}

else

{

Console.WriteLine(number + " NOT an Armstrong number");

result = 0;

Console.ReadLine();

}

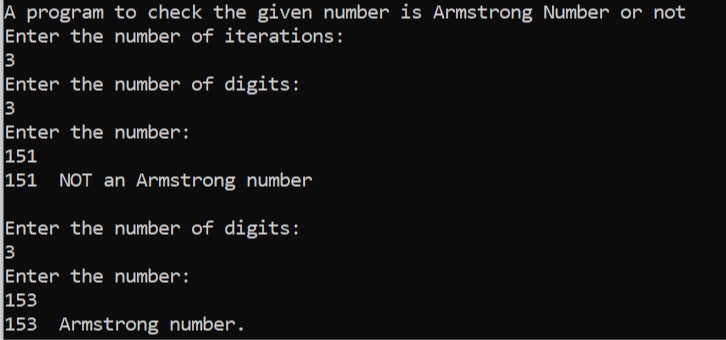
}

}

}

}

**Output:-**



**2.** **Write a program to print factorial of a number**

using System;

namespace myproject

{

public class Factorial

{

public static void Main(string[] args)

{

Console.WriteLine("A program to print factorial of the given number n");

int m, n, fact = 1;

Console.WriteLine("Enter the value of n: ");

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

for (m = 1; m <= n; m++)

{

fact = fact \* m;

}

Console.WriteLine(fact);

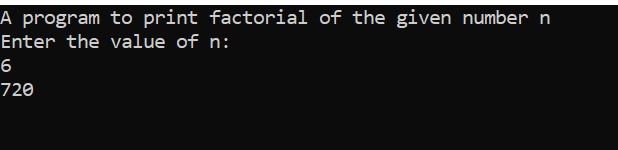
Console.ReadLine();

}

}

}

**Output:-**



**3.** **Write a program to find the GCD of two numbers**

using System;

namespace myproject

{

public class GCD

{

public static void Main(string[] args)

{

int n, a, b, gcd = 1, temp;

Console.WriteLine("A program to find out the GCD of two numbers");

Console.WriteLine("Enter the number of iterations: ");

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

for (int i = 0; i < n; i++)

{

Console.WriteLine("Enter the no.:");

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

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

for(int z = 1; z<=a && z <= b; z++)

{

if (a%z==0 && b%z==0)

gcd = z;

}

while (b != 0)

{

temp = b;

b = a % b;

a = temp;

}

gcd = a;

Console.WriteLine("GCD = " + gcd);

Console.ReadLine();

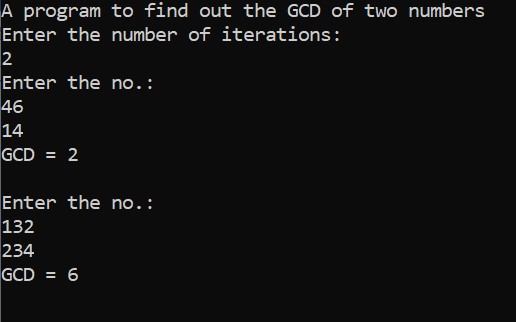
}

}

}

}

**Output:-**



**4. Write a program to check if a number is prime number**

using System;

namespace myproject

{

public class Prime

{

public static void Main(string[] args)

{

Console.WriteLine("A program to check the number is prime or not.");

int m, n, count = 0, i, j;

Console.WriteLine("Enter the number of iterations: ");

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

for (i = 0; i < j; i++)

{

Console.WriteLine("Enter the number: ");

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

for (m = 2; m <= n / 2; m++)

{

if (n % m == 0)

{

count = count + 1;

}

}

if (count == 0)

{

Console.WriteLine("This number " + n + " is a prime number");

}

else

{

Console.WriteLine("This number " + n + " is NOT a prime number");

count = 0;

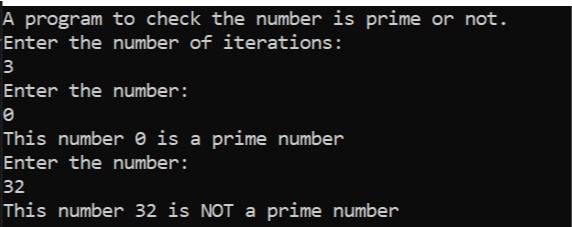
}

}

}

}

**Output :-**

****

**5. Write a program to print the Fibonacci series**

using System;

namespace myproject

{

public class Fibonacci

{

public static void Main(string[] args)

{

Console.WriteLine("A program to print fibonacci series of n terms");

int m, n, a = 0, b = 1;

Console.WriteLine("Enter the value: ");

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

Console.WriteLine("Fibonacci Serise: ");

for (m = 1; m <= n; m++)

{

Console.WriteLine(a);

int next = a + b;

a = b;

b = next;

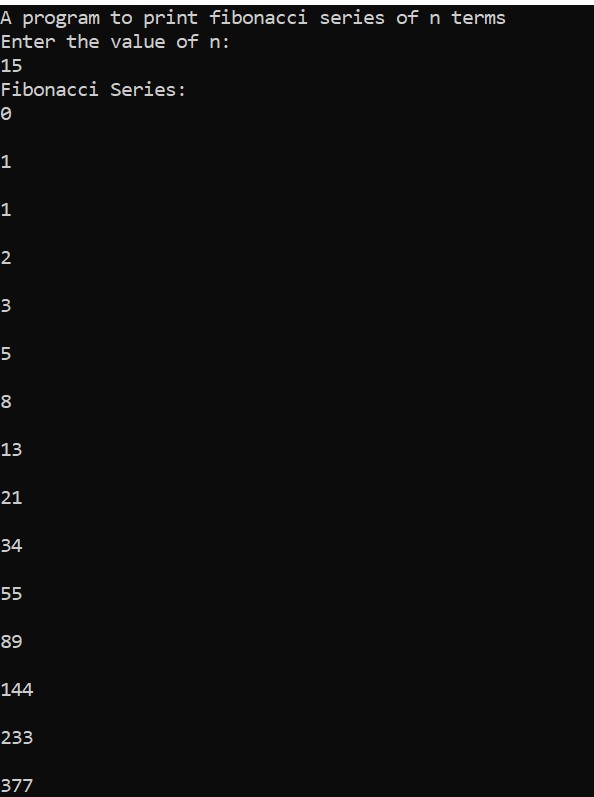
}

}

}

}

**Output:-**

****

**6. Write a program to print the half pyramid pattern**

using System;

namespace myproject

{

public class Pattern\_HalfPyramid

{

public static void Main(string[] args)

{

int space, rows;

Console.WriteLine("A program to print half pyramid pattern");

Console.WriteLine("Enter the number of rows:");

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

for (int i = 0; i <= rows; i++)

{

for (int star = 0; star < i; star++)

{

Console.Write("\*");

}

for (space = i; space < rows; space++)

{

Console.Write(" ");

}

Console.WriteLine();

Console.ReadLine();

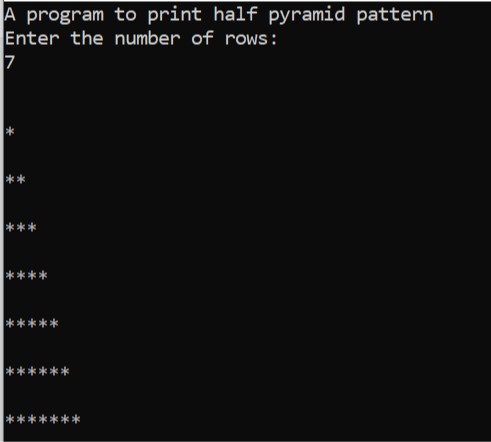
}

}

}

}

**Output:-**

****

**7. Write a program to print the half pyramid pattern with numbers**

using System;

namespace myproject

{

public class Pattern\_HalfPyramidNum

{

public static void Main(string[] args)

{

int space, rows;

Console.WriteLine("A program to print half pyramid pattern of numbers:");

Console.WriteLine("Enter the number of rows:");

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

Console.WriteLine();

for (int i = 1; i <= rows; i++)

{

for (int num = 1; num <= i; num++)

{

Console.Write(num);

}

for (space = i; space < rows; space++)

{

Console.Write(" ");

}

Console.WriteLine();

Console.ReadLine();

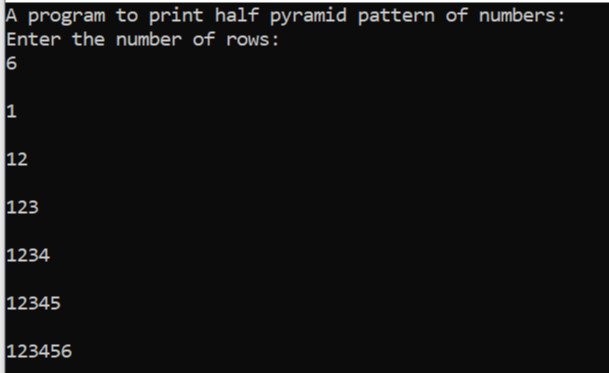
}

}

}

}

**Output :-**

****

**8. Write a program to print the half pyramid inverse pattern**

using System;

namespace myproject

{

public class Pattern\_HalfInversePyramid

{

public static void Main(string[] args)

{

int space, rows;

Console.WriteLine("A program to print half pyramid inverse pattern");

Console.WriteLine("Enter the number of rows:");

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

Console.WriteLine();

for (int i = 0; i <= rows; i++)

{

for (int star = rows; star > i; star--)

{

Console.Write("\*");

}

for (space = i; space < rows; space++)

{

Console.Write(" ");

}

Console.WriteLine();

Console.ReadLine();

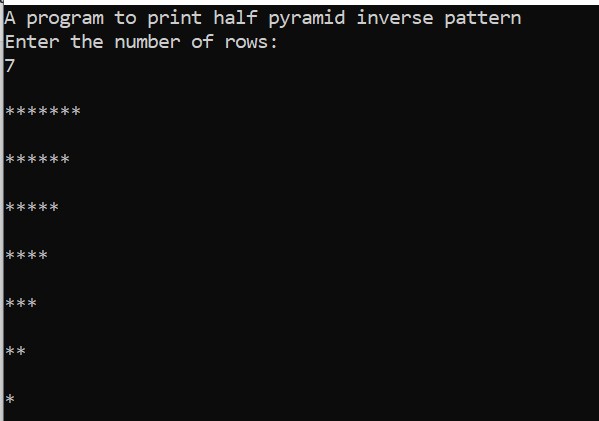
}

}

}

}

**Output:-**

****

**9. Write a program to print the pyramid pattern**

using System;

namespace myproject

{

public class Pattern\_pyramid

{

public static void Main(string[] args)

{

int space, rows;

Console.WriteLine("A program to print pyramid pattern");

Console.WriteLine("Enter the number of rows:");

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

for (int i = 1; i <= rows; i++)

{

for (space = i; space < rows; space++)

{

Console.Write(" ");

}

for (int star = 1; star < (i \* 2); star++)

{

Console.Write("\*");

}

Console.WriteLine();

Console.ReadLine();

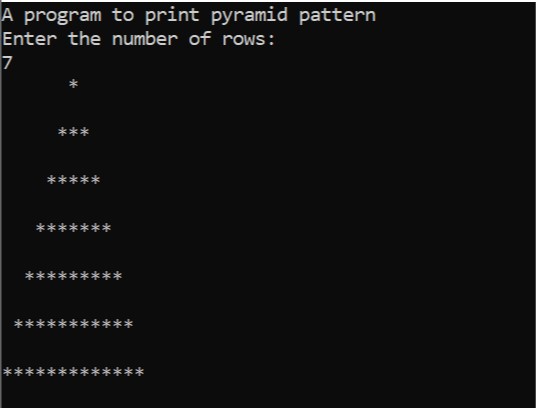
}

}

}

}

**Output:-**

****

**10. Write a program to print the inverse pyramid pattern**

using System;

namespace myproject

{

public class Pattern\_PyramidInverse

{

public static void Main(string[] args)

{

int space, rows;

Console.WriteLine("A program to print inverse pyramid pattern");

Console.WriteLine("Enter the number of rows:");

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

for (int i = rows; i >= 1; i--)

{

for (space = i; space <= rows; space++)

{

Console.Write(" ");

}

for (int star = (i \* 2); star > 1; star--)

{

Console.Write("\*");

}

Console.WriteLine();

Console.ReadLine();

}

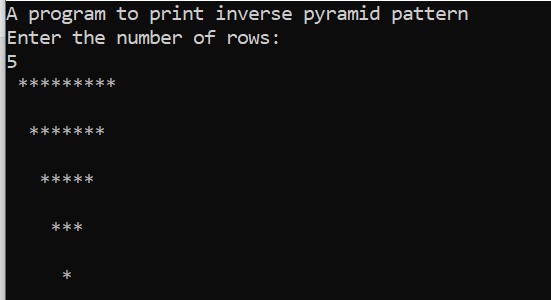
}

}

}

}

**Output:-**

****

**11. Write a program to print the diamond pattern**

using System;

namespace myproject

{

public class Pattern\_diamond

{

public static void Main(string[] args)

{

int rows, i, j, space;

Console.Write("Enter the number of rows: ");

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

for (i = 0; i <= rows / 2; i++)

{

for (space = i; space < rows / 2; space++)

{

Console.Write(" ");

}

for (j = 0; j <= i \* 2; j++)

{

Console.Write("\*");

}

Console.WriteLine();

}

for (i = rows / 2 + 1; i >= 1; i--)

{

for (space = i; space <= rows / 2 + 1; space++)

{

Console.Write(" ");

}

for (j = i \* 2 - 4; j >= 0; j--)

{

Console.Write("\*");

}

Console.WriteLine();

Console.ReadLine();

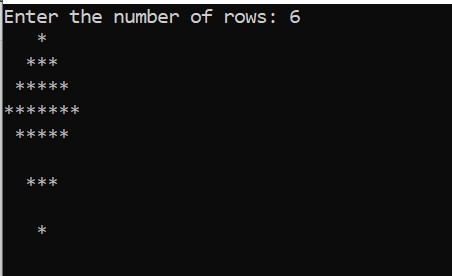
}

}

}

}

**Output:-**

****

**12. Write a program to print the Pascal’s triangle**

using System;

namespace myproject

{

public class Pascal\_Triangle

{

public static int Factorial(int fact)

{

int m, f = 1;

for (m = 1; m <= fact; m++)

{

f = f \* m;

}

return f;

}

public static int Ncr(int a, int b)

{

return Factorial(a) / (Factorial(b) \* Factorial(a - b));

}

public static void Main(string[] args)

{

int space, rows, c;

Console.WriteLine("A program to print the Pascal triangle.");

Console.WriteLine("Enter the number of rows:");

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

for (int i = 0; i <= rows; i++)

{

for (space = i; space < rows; space++)

{

Console.Write(" ");

}

for (int j = 0; j <= i; j++)

{

c = Ncr(i, j);

Console.Write(c + " ");

}

Console.WriteLine();

Console.ReadLine();

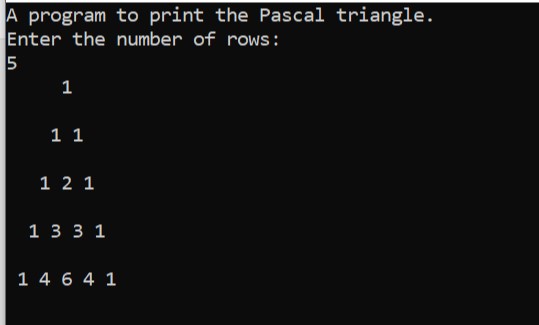
}

}

}

}

**Output:-**

****

**13. Write a program to compare two string without using string library functions**

using System;

namespace String

{

public class StringCompare

{

public static void Main(string[] args)

{

string str1, str2;

int flag = 0;

Console.WriteLine("A program to compare two

strings.");

Console.WriteLine("Enter string 1: ");

str1 = Console.ReadLine();

Console.WriteLine("Enter string 2: ");

str2 = Console.ReadLine();

for (int i = 0; i < str1.Length; i++)

{

if (str1[i] != str2[i])

{

flag = 0;

break;

}

else

{

flag = 1;

}

}

if (flag == 0)

{

Console.WriteLine(str1 + " and " + str2 + " are

NOT equal");

}

else if (flag == 1)

{

Console.WriteLine(str1 + " and " + str2 + " are Equal");

}

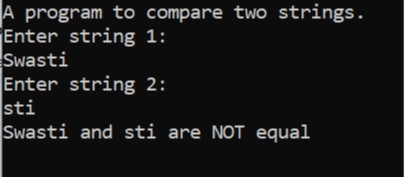
Console.readline();

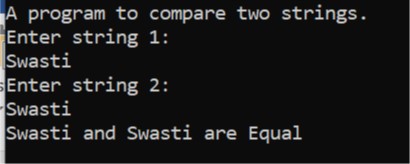
}

}

}

**Output:-**

****

****

**14. Write a program to count a total number of alphabets, digits and special characters in a string**

using System;

namespace String

{

public class StringCount

{

public static void Main(string[] args)

{

string str;

int alpha = 0, digit = 0, sym = 0;

Console.WriteLine("Enter the main string: ");

str = Console.ReadLine();

foreach (char s in str)

{

if (s >= 65 && s <= 90 || s >= 97 && s <= 122)

{

alpha += 1;

}

else if (s >= 48 && s <= 57)

{

digit += 1;

}

else

{

sym += 1;

}

}

Console.WriteLine();

Console.WriteLine("Number of Alphabets: " + alpha);

Console.WriteLine("Number of Digits: " + digit);

Console.WriteLine("Number of Special Characters: " + sym);

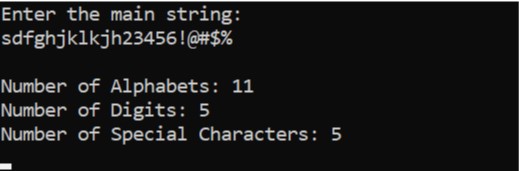
Console.ReadLine();

}

}

}

**Output:-**

****

**15. Write a program to copy one string to another string**

using System;

namespace String

{

public class StringCopy

{

public static void Main(string[] args)

{

string s1, s2 = "Not Copied";

Console.WriteLine("A program to copy one string to another string");

Console.WriteLine("Enter the string: ");

s1 = Console.ReadLine();

foreach (char a in s1)

{

s2 = s1;

}

Console.WriteLine("String Copied");

Console.WriteLine("Copied String is: " + s2);

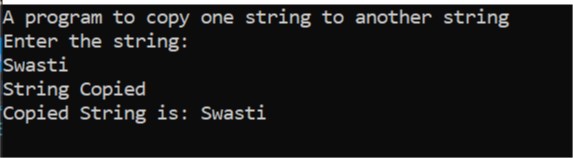
Console.ReadLine();

}

}

}

**Output:-**

****

**16. Write a program to find maximum occurring character in a string**

using System;

namespace Strings

{

public class StringMax

{

public static void Main(string[] args)

{

string str;

int[] count = new int[256];

Console.WriteLine("Enter the string: ");

str = Console.ReadLine();

for (int i = 0; i < str.Length; i++)

{

count[str[i]]++;

}

int max = -1;

char result = ' ';

for (int i = 0; i < str.Length; i++)

{

if (max < count[str[i]])

{

max = count[str[i]];

result = str[i];

}

}

Console.WriteLine("Maximum occuring character in string: " + result);

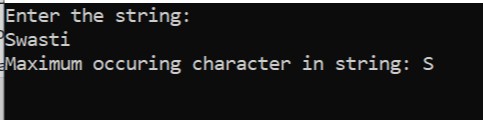
Console.ReadLine();

}

}

}

**Output:**

****

**17. Write a program to check whether a given substring is present in the given string**

using System;

namespace Strings

{

public class StringSubString

{

public static void Main(string[] args)

{

string str, substr;

Console.WriteLine("A program to check whether a given substring is present in the given string.");

Console.WriteLine("Enter the string: ");

str = Console.ReadLine();

Console.WriteLine("Enter the Sub-string");

substr = Console.ReadLine();

int flag = 0;

for (int i = 0; i <= str.Length - substr.Length; i++)

{

for (int j = i; j < i + substr.Length; j++)

{

flag = 1;

if (str[j] != substr[j - i])

{

flag = 0;

break;

}

}

if (flag == 1)

break;

}

if (flag == 1)

{

Console.WriteLine("The substring is present in given String");

}

else

{

Console.WriteLine("The substring is NOT present in given String");

Console.ReadLine();

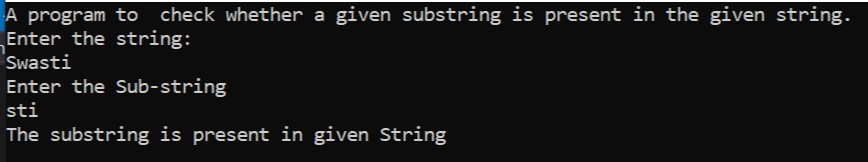
}

}

}

}

**Output:**

****

**18. Write a Program for Encapsulation**

using System;

namespace test

{

class Student

{

private int roll;

private string name;

public int Roll

{

get

{

return roll;

}

set

{

roll = value;

}

}

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

}

class Program

{

static void Main(string[] args)

{

Student A = new Student();

A.Roll = 21;

A.Name = "Swasti";

Console.WriteLine("Roll: " + A.Roll);

Console.WriteLine("Name: " + A.Name);

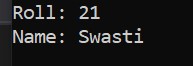
Console.ReadLine();

}

}

}

**Output:**

****

**19. Write a program for Abstraction**

using System;

namespace OOPs

{

public class Abstraction

{

abstract class Cs

{

public abstract void Fun();

}

private class Good : Cs

{

public override void Fun()

{

Console.WriteLine("C# is Good");

}

}

private class Best : Cs

{

public override void Fun()

{

Console.WriteLine("C# is Best");

}

}

private class Better : Cs

{

public override void Fun()

{

Console.WriteLine("C# is Better");

}

}

public class MyClass

{

public static void Main()

{

Cs c;

c = new Good();

c.Fun();

c = new Best();

c.Fun();

c = new Better();

c.Fun();

Console.ReadLine();

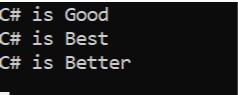
}

}

}

}

**Output:**

****

**20. Write a program for single Inheritence**

using System;

namespace OOPs

{

public class Inheritence

{

class MainClass

{

public void Print()

{

Console.WriteLine("Print");

}

}

class Subclass : MainClass

{

void Print1()

{

Console.WriteLine("Print 2");

}

static void Main(string[] args)

{

Subclass s = new Subclass();

s.Print();

s.Print1();

Console.ReadLine();

}

}

}

}

**Output:**



**21. Write a program for Multilevel Inheritence**

using System;

namespace OOPs

{

public class MultiInheritence

{

class MainClass

{

public void Print()

{

Console.WriteLine("Print");

}

}

class Subclass : MainClass

{

public void Print1()

{

Console.WriteLine("Print 1");

}

}

class Subclass2 : Subclass

{

public void Print2()

{

Console.WriteLine("Print 2");

}

static void Main(string[] args)

{

Subclass2 s = new Subclass2();

s.Print();

s.Print1();

s.Print2();

Console.ReadLine();

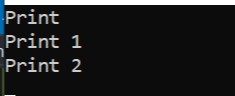
}

}

}

}

**Output:**

****

**22. Write a program for multiple Inheritence**

using System;

namespace OOPs

{

public class MultipleInheritence

{

class MainClass

{

public void Print()

{

Console.WriteLine("Print");

}

}

interface MainClass1

{

void Print1();

}

class Subclass : MainClass, MainClass1

{

void Print2()

{

Console.WriteLine("Print 2");

}

public void Print1()

{

Console.WriteLine("Print 1");

}

static void Main(string[] args)

{

Subclass s = new Subclass();

s.Print();

s.Print1();

s.Print2();

Console.ReadLine();

}

}

}

}

**Output:**

****

**23. Write a program for method overloading**

using System;

namespace OOPs

{

public class MethodOverloading

{

static int Sum(int a, int b)

{

return a + b;

}

static double Sum(double a, double b)

{

return a + b;

}

public static void Main()

{

int sum1 = Sum(73, 34);

double sum2 = Sum(75.84, 74.16);

Console.WriteLine(sum1);

Console.WriteLine(sum2);

Console.ReadLine();

}

}

}

**Output:**

****

**24. Write a program for method overriding**

using System;

namespace OOPs

{

public class MethodOverriding

{

public class Cs

{

public virtual void Fun()

{

Console.WriteLine("B.Sc");

}

}

public class MCA : Cs

{

public override void Fun()

{

Console.WriteLine("MCA");

}

}

private class BCA : Cs

{

public override void Fun()

{

Console.WriteLine("BCA");

}

}

public static void Main()

{

Cs c;

c = new BCA();

c.Fun();

c = new MCA();

c.Fun();

Console.ReadLine();

}

}

}

**Output:**

****

**25. Write a program for Interface**

using System;

namespace OOPs

{

public class Interface

{

public interface Cs

{

void Fun();

}

private class Bca : Cs

{

public void Fun()

{

Console.WriteLine("BCA");

}

}

private class Mca : Cs

{

public void Fun()

{

Console.WriteLine("MCA");

}

}

public class MyClass

{

public static void Main(string[] args)

{

Cs c;

c = new Bca();

c.Fun();

c = new Mca();

c.Fun();

Console.ReadLine();

}

}

}

}

**Output:**

****

**26. Write a program for Namespace**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

A.B test = new A.B();

test.C();

}

}

}

namespace A

{

public class B

{

public void C()

{

Console.WriteLine("Hi");

Console.ReadLine();

}

}

}

**Output:**

****

**27. Write a program for exception handling through try and catch**

using System;

namespace ConsoleApp

{

class Program

{

static void Main(string[] args)

{

int[] myNumbers = { 1, 2, 3,4 };

try

{

Console.WriteLine(myNumbers[5]);

}

catch (Exception c)

{

Console.WriteLine(c.Message);

Console.ReadLine();

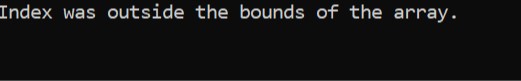
}

}

}

}

**Output:**

****

**28. Write a program for Properties**

using System;

namespace OOPBasics

{

public class Properties

{

public class CSharp

{

public int roll;

public string name;

private string Result;

public CSharp(int a, string b, string c)

{

roll = a;

name = b;

Result1 = c;

}

public string Result1

{

get

{

return Result;

}

set

{

if (value == "good" || value == "average" || value == "bad")

{

Result = value;

}

else

{

Result = "Not Valid";

}

}

}

}

}

class Program

{

static void Main(string[] args)

{

Properties.CSharp c1 = new Properties.CSharp(1, "Xyz", "good");

Properties.CSharp c2 = new Properties.CSharp(2, "Ijk", "5");

Console.WriteLine(c1.roll + " " + c1.name + " " + c1.Result1);

Console.WriteLine(c2.roll + " " + c2.name + " " + c2.Result1);

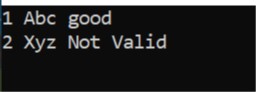
Console.ReadLine();

}

}

}

**Output:**

****

**29. Write a program for Constructors**

using System;

using OOPBasics;

namespace OOPBasics

{

class Constructor

{

public class CSharp

{

public int roll;

public string name;

public int marks;

public CSharp(int a, string b, int c)

{

roll = a;

name = b;

marks = c;

}

}

}

public class MyClass

{

public static void Main(string[] args)

{

Constructor.CSharp a = new Constructor.CSharp(21, "Swasti", 7);

Console.WriteLine("Roll: " + a.roll + "\nName: " + a.name + "\nMarks: " + a.marks);

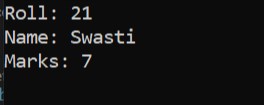
Console.ReadLine();

}

}

}

**Output:**

****

**30. Write a program for Threading**

using System;

using System.Threading;

namespace Threading

{

class Program

{

static void T1()

{

Console.WriteLine("Thread1 Started");

Thread.Sleep(5000);

Console.WriteLine("Thread1 executing");

Thread.Sleep(5000);

Console.WriteLine("Thread1 executing");

}

static void T2()

{

Console.WriteLine("Thread2 Started");

Thread.Sleep(5000);

Console.WriteLine("Thread2 executing");

Thread.Sleep(5000);

Console.WriteLine("Thread2 executing");

}

public static void Main()

{

Thread t1 = new Thread(T1);

Thread t2 = new Thread(T2);

t1.Start();

t2.Start();

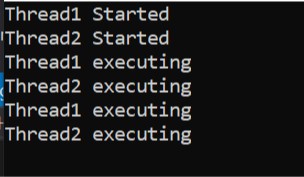
Console.ReadLine();

}

}

}

**Output:**

****

**31. Write a program for Indexer**

using System;

class IndexerCreation

{

private string[] val = new string[4];

public string this[int index]

{

get

{

return val[index];

}

set

{

val[index] = value;

}

}

}

class MyClass

{

public static void Main()

{

IndexerCreation ic = new IndexerCreation();

ic[0] = "Amisha";

ic[1] = "Gaurvi";

ic[2] = "Shail";

ic[3] = "Swasti";

Console.Write("Printing values stored in objects used as arrays\n");

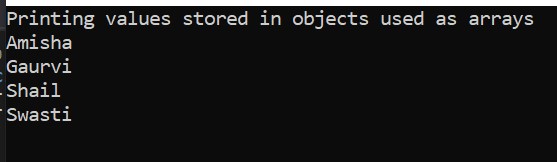
Console.WriteLine(ic[0] + "\n" + ic[1] + "\n" + ic[2] + "\n" + ic[3]);

Console.ReadLine();

}

}

**Output:**

****

**32. Write a program to access data from database using ADO.NET**

using System;

using System.Data;

using System.Data.SqlClient;

namespace test

{

class Program

{

public static void Main(string[] args)

{

string connectionString;

SqlConnection conn;

connectionString = @"Data Source=DESKTOP-VPKMCMC\SQLEXPRESS;Initial Catalog=demo;Trusted\_Connection=true";

conn = new SqlConnection(connectionString);

conn.Open();

Console.WriteLine("Connected to Database!");

string query = "select \* from student";

SqlCommand cmd = new SqlCommand(query, conn);

SqlDataReader dataReader = cmd.ExecuteReader();

Console.WriteLine(dataReader.GetName(0) + " "+dataReader.GetName(1)+" "+dataReader.GetName(2));

while (dataReader.Read())

{

Console.WriteLine(dataReader.GetValue(0) + " " +

dataReader.GetValue(1) + " " + dataReader.GetValue(2));

}

Console.ReadLine();

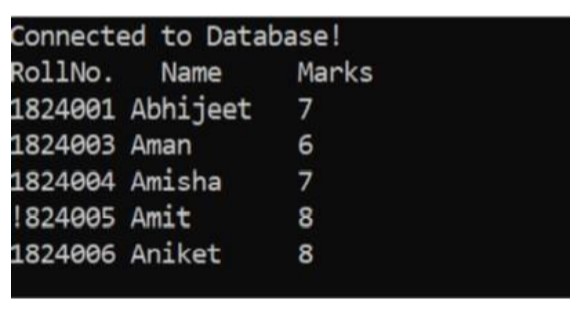
conn.Close();

}

}

}

**Output:**

****