

**DPG Institute of Technology & Management**

**Sector -34, Gurugram**

**WEB DESIGNING**

**&**

**Object-Oriented Programming**

**PRACTICAL FILE**

**Submitted to :**

**Ms Rashmi verma (oops)**

**Ms Monika(WD)**

**Submitted By :**

**Name -kishan kumar ID -110712**

**Course -BCA 3rd Sem. Section - “A”**

# WD INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO** | **PRACTICAL** | **DATE** | **SIGNATURE** |
| 1 | **Write a program for basic structure of an HTML document** |  |  |
| 2 | **Write a program to show the effect of various heading tags** |  |  |
| 3 | **Write a program to create a basic table** |  |  |
| 4 | **Write a program to link tags** |  |  |
| 5 | **Write a program to design a ordered and unordered list** |  |  |
| 6 | **Make a Project to create a simple website using HTML & CSS** |  |  |

OOPS-INDEX

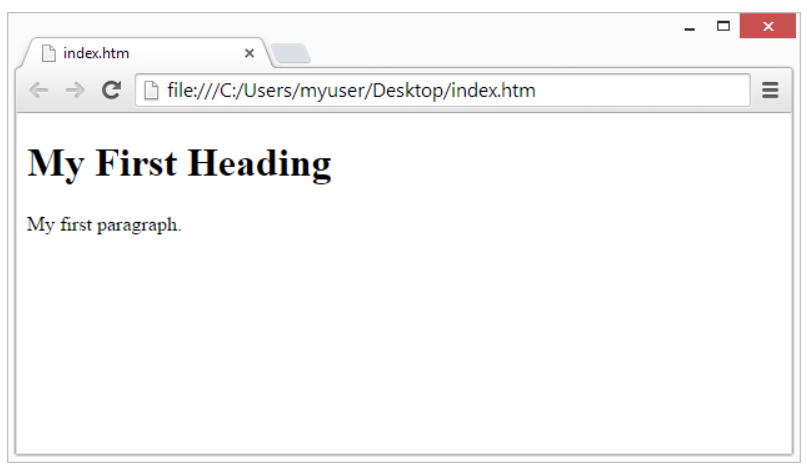
|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO.** | **PRACTICAL** | **DATE** | **SIGNATURE** |
| **1** | **Write a program to create a class for complex no write function to get values int add two complex no.** |  |  |
| **2** | Write a program to find factorial of a number using recursion |  |  |
| **3** | **Write a program to implement constructor and destructor using new & delete operator** |  |  |
| **4** | **Write a program to create a calculator using switch case.** |  |  |
| **5** | **Write a program to implement virtual function and function overriding using inheritance** |  |  |
| **6** | **Write a program to swapping of two numbers using pointer** |  |  |

# Web Designing

## PRACTICAL :- 1

**Write a program for basic structure of an HTML document**

<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>



**PRACTICAL :-2**

**Write a program to show the effect of various heading tags**

<html>

<head>

<title>DataFlair</title>

</head>

<body>

<h1>DataFlair Heading 1</h1>

<h2>DataFlair Heading 2</h2>

<h3>DataFlair Heading 3</h3>

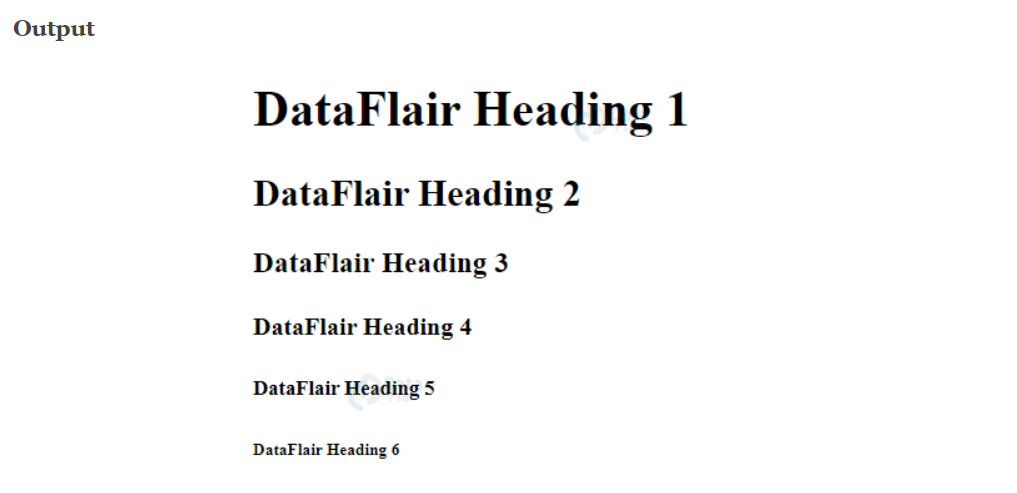
<h4>DataFlair Heading 4</h4>

<h5>DataFlair Heading 5</h5>

<h6>DataFlair Heading 6</h6>

</body>

</html>



**PRACTICAL :-3**

**Write a program to create a basic table.**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<h1>Programming Languages</h1>

<table>

<tr>

<th>Language</th>

<th>Release Year</th>

</tr>

<tr>

<td>Java</td>

<td>1995</td>

</tr>

<tr>

<td>Pascal</td>

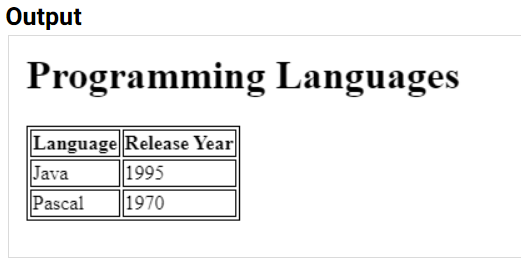
<td>1970</td>

</tr>

</table>

</body>

</html>



**PRACTICAL : -4**

**Write a program to link tags.**

<!DOCTYPE html>

<html>

<h3>Example Of a link</h3>

<body>

<p>Click on the following link</p>

<ul>

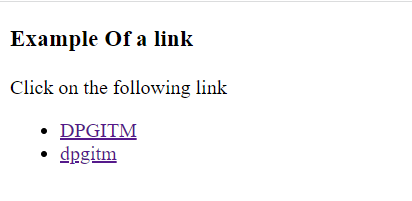
<li><a href = "https://www.dpgitm.ac.in/">DPGITM</a> </li>

<li><a href = "https://www.dpgitm.ac.in/">dpgitm</a> </li>

</ul>

</body>

</html>



## PRACTICAL :- 5

**Write a program to design a ordered and unordered list.**

**UNORDERED LIST**

<!DOCTYPE html>

<html>

<head>

<title>Title of the document</title>

</head>

<body>

<ul type="circle">

<li>List item </li>

<li>List item</li>

<li>List item</li>

</ul>

<ul type="square">

<li>List item</li>

<li>List item</li>

<li>List item</li>

</ul>

</body>

</html>



**ORDERED LIST**

<!DOCTYPE html>

<html>

<head>

<title>Title of the document</title>

</head>

<body>

<ol>

<li>List item </li>

<li>List item</li>

<li>List item</li>

</ol>

<ol>

<li>List item</li>

<li>List item</li>

<li>List item</li>

</ol>

</body>

</html>



## PRACTICAL :-6

**Make a Project to create a simple website using HTML & CSS**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>jai shri shyam</title>

<link href="https://fonts.googleapis.com/css2?family=Anton&family=League+Gothic&family=Pacifico&display=swap"

rel="stylesheet">

<style>

body {

padding: 0px;

margin: 0px;

background: black;

}

.left {

position: absolute;

/\* border: 2px solid rgb(212, 18, 18); \*/

display: inline-block;

left: 13px;

top: 22px;

margin: 3px;

padding: 2px;

cursor: pointer;

}

.left img {

width: 147px;

height: 62px;

text-align: center;

cursor: pointer;

}

.left div {

color: rgb(115 7 75);

font-family: 'Pacifico', cursive;

font-size: 25px;

margin: 2px;

padding: 0px;

}

.mid {

font-family: 'Pacifico', cursive;

/\* border: 2px solid rgb(25, 119, 64); \*/

display: block;

width: 50%;

margin: 2px auto;

padding: 2px;

}

.right {

font-family: 'Pacifico', cursive;

position: absolute;

right: 35px;

top: 31px;

/\* border: 2px solid rgb(34, 10, 78); \*/

display: inline-block;

padding: 2px;

margin: 2px;

}

.navbar {

display: inline-block;

}

.navbar li {

display: inline-block;

padding: 3px;

margin: 13px;

}

.navbar li a {

font-size: 25px;

color: rgb(184, 17, 81);

text-decoration: none;

}

.navbar li a:hover {

color: rgb(167, 107, 162);

text-decoration: underline;

}

.navbar li a :active {

color: rgb(23, 233, 233);

text-decoration: underline;

}

.navbar li a :visited {

color: rgb(82, 219, 219);

text-decoration: underline;

}

.right button {

color: white;

background-color: black;

border-radius: 4px;

}

.story {

/\* border: 2px solid black; \*/

/\* display: inline-block; \*/

width: 500px;

height: 777px;

/\* left: 50px; \*/

/\* top: 300px; \*/

margin: 72px 15px;

padding: 30px;

}

.story h1{

color:#9d2727;

}

.story p{

color: white;

}

</style>

</head>

<body>

<header>

<div class="left">

<img src="dpgitm.jpg" alt="img not loaded">

<div> DPGITM</div>

</div>

<div class="mid">

<ul class="navbar">

<li><a target="\_blank"

href="https://search.yahoo.com/search?fr=mcafee&type=E211US714G91657&p=facebook">Home</a></li>

<li><a target="\_blank" href="https://shrishyamdarshan.in/darshan-booking/">Booking</a> </li>

<li><a href="#">About</a></li>

<li><a href="#">Contactdetails</a></li>

<li><a href="#">Bookingdetails</a></li>

</ul>

</div>

<div class="right">

<button>contactus</button>

<button>help</button>

</div>

</header>

<div class="story">

<h1>DPGITM</h1>

<p>

DPG Group was established in 2004 on a sprawling campus spread over 23 acres.. The Group had set up its first institute as DPG Chaudhary Partap Singh Memorial Trust (CPSM) for B.Ed. & D.Ed. The institute is approved by NCTE, Jaipur and affiliated to MDU, Rohtak. With three years of excellence in the field of education, this group in 2007 went ahead to open second institute as DPG Polytechnic to award diploma to the students in Gurgaon. Having established itself and providing quality education, the group came up with DPGITM Engineering College in 2011, affiliated to MDU, Rohtak, providing technical education to students in Haryana. In 2015, this group introduced DPG Degree College.

The Trust members of DPG Institute of Technology and Management (DPGITM) are academicians and seasoned professionals with an experience of almost two decades in the field of higher education in India. DPGITM campus is located in Gurgaon, the district Head-quarter of Haryana and one of the major Corporate Hub of the Northern Capital Region (Delhi-NCR).

DPG Institute of Technology and Management is approved by AICTE , Ministry of HRD, Govt. of India; Director of Technical Education, Govt. of Haryana and Affiliated to MD University, Rohtak. The Governing body of the Institutes comprise of Entrepreneurs, Industrialists, Academicians and Professionals who have achieved much success in their respective fields.

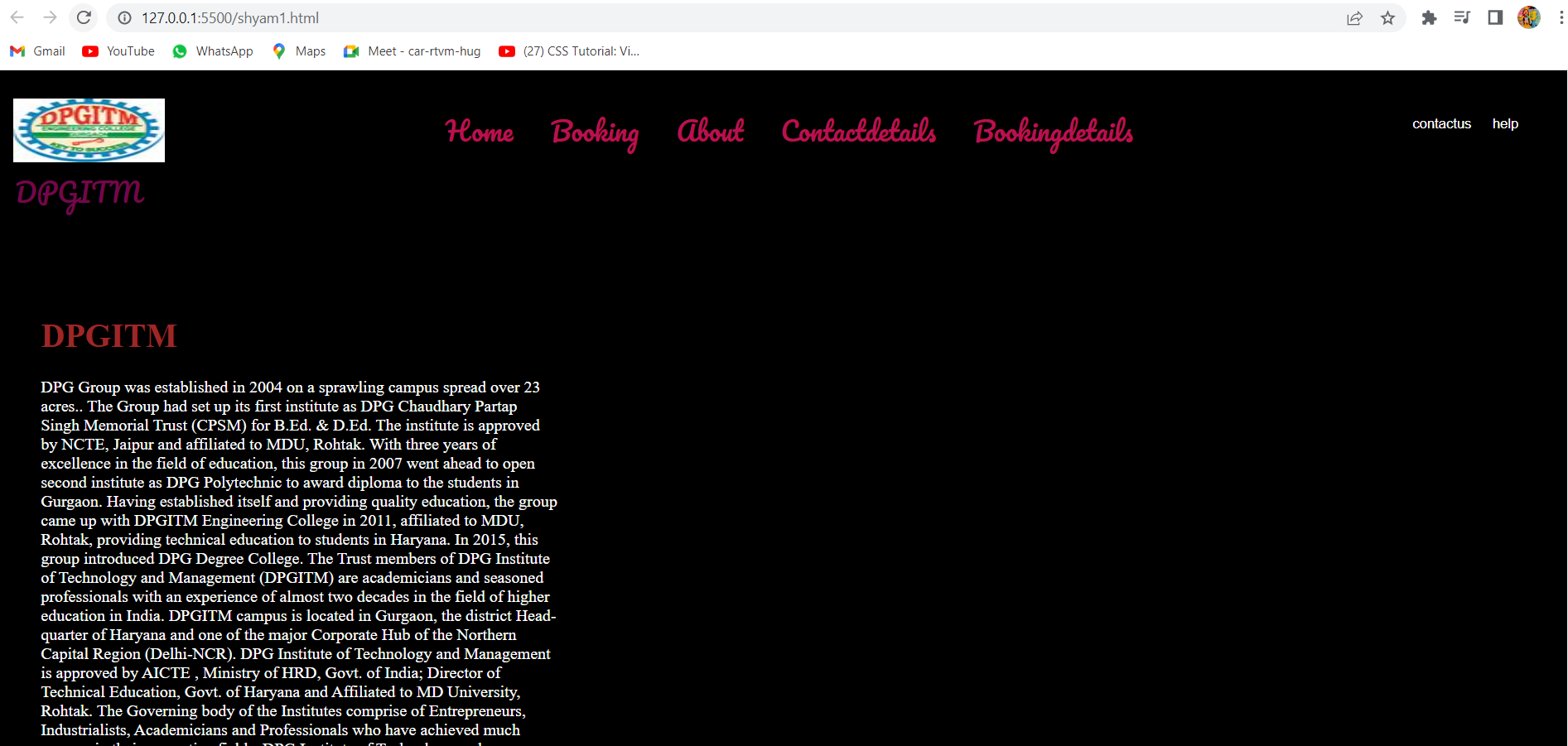
DPG Institute of Technology and Management is one of the best managed self-financed Colleges of Haryana, imparting quality technical education using most modern techniques. The institute gives top priorities to interactive dialogue with students on one to one basis. The ongoing online interaction between students and teachers as well as confidential feed back from the students on each course in every semester is the hallmark of DPG Institute of Technology and Management Education System. The institute ensures that every student is skilled in all aspects when they are moved to work in the industry. It had an intake capacity of 300 seats in CSE, ME, CE, EE, ECE (60 seats for each program).

</p>

</div>

</body>

</html>



**Object-Oriented Programming**

**PRACTICAL :-1**

**Write a program to create a class for complex no write function to get values int add two complex no.**

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

**using** **namespace** std;

// User Defined Complex class

**class** Complex {

    // Declaring variables

**public**:

**int** real, imaginary;

    // Constructor to accept

    // real and imaginary part

    Complex(**int** tempReal = 0, **int** tempImaginary = 0)

    {

        real = tempReal;

        imaginary = tempImaginary;

    }

    // Defining addComp() method

    // for adding two complex number

    Complex addComp(Complex C1, Complex C2)

    {

        // creating temporary variable

        Complex temp;

        // adding real part of complex numbers

        temp.real = C1.real + C2.real;

        // adding Imaginary part of complex numbers

        temp.imaginary = C1.imaginary + C2.imaginary;

        // returning the sum

**return** temp;

    }

};

// Main Class

**int** main()

{

    // First Complex number

    Complex C1(3, 2);

    // printing first complex number

    cout<<"Complex number 1 : "<< C1.real

                << " + i"<< C1.imaginary<<endl;

    // Second Complex number

    Complex C2(9, 5);

    // printing second complex number

    cout<<"Complex number 2 : "<< C2.real

                << " + i"<< C2.imaginary<<endl;

    // for Storing the sum

    Complex C3;

    // calling addComp() method

    C3 = C3.addComp(C1, C2);

    // printing the sum

    cout<<"Sum of complex number : "

                       << C3.real << " + i"

                       << C3.imaginary;

}

// Importing all libraries

#include<bits/stdc++.h>

using namespace std;

// User Defined Complex class

class Complex {

// Declaring variables

public:

int real, imaginary;

// Constructor to accept

// real and imaginary part

Complex(int tempReal = 0, int tempImaginary = 0)

{

real = tempReal;

imaginary = tempImaginary;

}

// Defining addComp() method

// for adding two complex number

Complex addComp(Complex C1, Complex C2)

{

// creating temporary variable

Complex temp;

// adding real part of complex numbers

temp.real = C1.real + C2.real;

// adding Imaginary part of complex numbers

temp.imaginary = C1.imaginary + C2.imaginary;

// returning the sum

return temp;

}

};

// Main Class

int main()

{

// First Complex number

Complex C1(3, 2);

// printing first complex number

cout<<"Complex number 1 : "<< C1.real

<< " + i"<< C1.imaginary<<endl;

// Second Complex number

Complex C2(9, 5);

// printing second complex number

cout<<"Complex number 2 : "<< C2.real

<< " + i"<< C2.imaginary<<endl;

// for Storing the sum

Complex C3;

// calling addComp() method

C3 = C3.addComp(C1, C2);

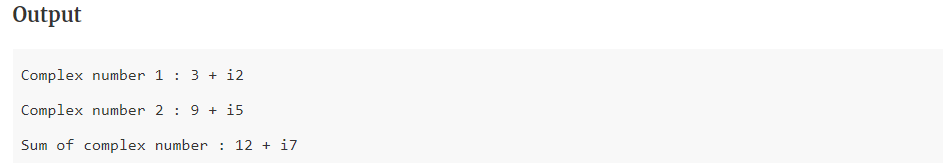
// printing the sum

cout<<"Sum of complex number : "

<< C3.real << " + i"

<< C3.imaginary;

}

****

## PRACTICAL :-2

**Write a program to find factorial of a number using recursion.**

#include<iostream>

**using** **namespace** std;

**int** main()

{

**int** factorial(**int**);

**int** fact,value;

cout<<"Enter any number: ";

cin>>value;

fact=factorial(value);

cout<<"Factorial of a number is: "<<fact<<endl;

**return** 0;

}

**int** factorial(**int** n)

{

**if**(n<0)

**return**(-1); /\*Wrong value\*/

**if**(n==0)

**return**(1);  /\*Terminating condition\*/

**else**

{

**return**(n\*factorial(n-1));

}

}



**PRACTICAL :-3**

**Write a program to implement constructor and destructor using new & delete operator.**

**PRACTICAL :-4**

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

# include <iostream>

using namespace std;

int main() {

char op;

float num1, num2;

cout << "Enter operator: +, -, \*, /: ";

cin >> op;

cout << "Enter two operands: ";

cin >> num1 >> num2;

switch(op) {

case '+':

cout << num1 << " + " << num2 << " = " << num1 + num2;

break;

case '-':

cout << num1 << " - " << num2 << " = " << num1 - num2;

break;

case '\*':

cout << num1 << " \* " << num2 << " = " << num1 \* num2;

break;

case '/':

cout << num1 << " / " << num2 << " = " << num1 / num2;

break;

default:

// If the operator is other than +, -, \* or /, error message is shown

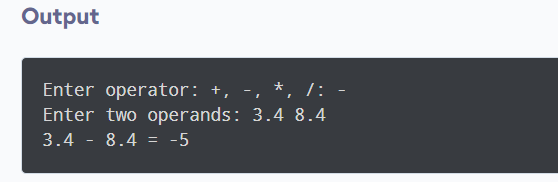
cout << "Error! operator is not correct";

break;

}

return 0;

}



**PRACTICAL :- 5**

**Write a program to implement virtual function and function overriding using inheritance.**

#include

using namespace std;

class b

{

public:

virtual void show()

{

cout<<"\n Showing base class....";

}

void display()

{

cout<<"\n Displaying base class...." ;

}

};

class d:public b

{

public:

void display()

{

cout<<"\n Displaying derived class....";

}

void show()

{

cout<<"\n Showing derived class....";

}

};

int main()

{

b B;

b \*ptr;

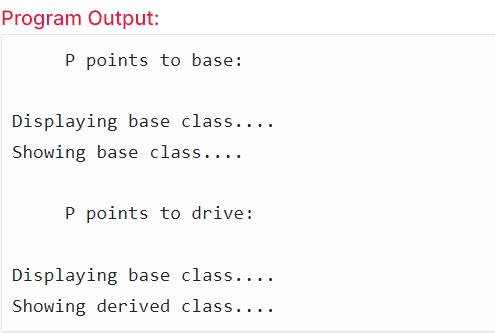
cout<<"\n\t P points to base:\n" ; ptr=&B; ptr->display();

ptr->show();

cout<<"\n\n\t P points to drive:\n"; d D; ptr=&D; ptr->display();

ptr->show();

}



**PRACTICAL :- 6**

**Write a program to swapping of two numbers using pointer.**

 #include <iostream>

**using** **namespace** std;

**int** main()

{

**int** a=5, b=10;

cout<<"Before swap a= "<<a<<" b= "<<b<<endl;

a=a\*b; //a=50 (5\*10)

b=a/b; //b=5 (50/10)

a=a/b; //a=10 (50/5)

cout<<"After swap a= "<<a<<" b= "<<b<<endl;

**return** 0;

}

