

S.No	Program Names	Pages	Remark	S.
1.	//WAP to print your Intro C++			
2.	//WAP to print the Value using variable.			
3.	//WAP to calculate sum & avg of three number.			
4.	//WAP to calculate area of circle, rectangle, triangle.			
5.	//WAP to print the basic detail of student using diff data type.			
6.	//WAP to use cin & cout by using user-defined function.			
7.	//WAP to use Enum data type in C++			
8.	//WAP to use new operator in C++ / delete			
9.	//WAP to use of structure in C++			
10.	//WAP to demonstrate use of reference variable.			
11.	//WAP to use scope resolution operator in C++			
12.	//WAP to accessing data members of class in C++			
13.	//WAP to adding two no in class			
14.	//WAP to show the concept of encapsulation using class in C++.			
15.	• //WAP to using multiple func. in class for mathematical operation.			
16.	//WAP to show the concept of abstraction using math.h header file without class			

S.No	Program Names	Pages	Remark
17.	//WAP to show the concept of abstraction using math header file in class format.		
18.	//WAP to show the concept of local class in C++.		
19.	//WAP to show the concept of nested class to use of resolution operator in C++		
20.	//WAP to print outside declaration of member function using class		
21.	//WAP to initialization of abstraction in C++		
22.	//WAP to show the concept of Inline function without class in C++.		
23.	//WAP to show the concept of Inline function with class in C++.		
24	//WAP to print Fibonacci series in C++		
25.	//WAP to demonstrate the concept of static data member in C++		
26.	//WAP to print Inside declaration of constructor using class		
27	//WAP to print Outside declaration of constructor using class		
28.	//WAP to Show the concept of default Constructor.		

S.No	Program List	Pages	Re
29.	//WAP to show the concept of Parameterised Constructor.		
30.	//WAP to show the concept of copy constructor.		
31.	//WAP to show the concept of constructor overloading.		
32.	//WAP to show the concept of dynamic Constructor.		
33.	//WAP to show the concept of destructor		
34.	<del>//WAP to show the concept of default destructor.</del>		
34.	//WAP to show the concept of default argument.		

## PROGRAM I

//WAP to Print your Intro In C++

```
#include <conio.h>
#include <iostream.h>
void main()
```

S

```
clrscr();
cout << "My Name Is Abhinav Kaushik" << endl;
cout << "I am 18th year Old Boy" << endl;
cout << "My father Name Is Anil Kumar Sharma" <<
```

endl;

```
cout << "He is a English lecturer" << endl;
cout << "My hobbies are Singing and many more" << endl;
```

```
cout << "like playing Cricket, badminton" << endl;
```

```
cout << "And motivate to all my followers.";
```

```
getch();
```

}

## Output -

My Name Is Abhinav Kaushik

I am 18<sup>th</sup> year old boy \*

My Father Name Is Anil Kumar Sharma

He is a English lecturer

My Hobbies are Singing and many more

like playing Cricket, Badminton

And motivate to all my followers.

## Program 2

//WAP to Print the Value Using Variable

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    float a, b, sum;
    cout << "enter the value of a";
    cin >> a;
    cout << "enter value is = " << a << endl;
    cout << "enter the value of b";
    cin >> b;
    cout << "enter the value of b = " << b << endl;
    sum = a+b;
    cout << "the sum of a+b= " << sum;
    getch();
}
```

3

## Output -

enter the value of a 5

enter value is = 5

enter the value of b 6

enter the value is = 6

the sum of a+b = 11

### Program 3

//WAP to calculate sum & avg of three numbers.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    float a,b,c,sum,avg;
    cout << "enter the value of abc";
    cin >> a >> b >> c;
    cout << "enter the value of abc = " << a << b << c << endl;
    sum = a + b + c;
    cout << "the sum of a+b+c = " << sum << endl;
    avg = sum / 2;
    cout << "the avg of given no = " << avg;
    getch();
}
```

3

Output -

enter the value of abc

2

7

1

enter the value of abc = 271

the sum of  $a+b+c = 10$

the avg of given no = 5

## Program 4

//WAP to calculate area of circle, rectangle, triangle.

```
# include <iostream.h>
# include <conio.h>
void main()
{
    clrscr();
    float rad = 3, area;
    area = (3.14 * rad * rad);
    cout << "area of circle is " << area;
    getch();
}
```

Output -

area of circle is 28.26

## Area of Rectangle

```
# include <iostream.h>
# include<conio.h>
void main()
{
    clrscr();
    int length, breadth, area;
    cout<<"area of rectangle \n";
    cout<< "\n enter the length of rectangle: ";
    cin>>length;
    cout<< "\n enter the breadth of rectangle: ";
    cin >> breadth;
    area = length * breadth;
    cout<< "\n area of rectangle: "<<area;
    getch();
}
```

Output -

area of rectangle

enter the length of rectangle: 5

enter the breadth of rectangle: 4

area of rectangle: 20

## Area of triangle

```
# include <iostream.h>
# include <conio.h>
void main()
{
    clrscr();
    float b,h,area;
    cout<<"enter base length of triangle: ";
    cin>>b;
    cout<<"enter height length of triangle: ";
    cin>>h;
    area = 0.5 * b * h;
    cout << "\n area = " << area;
    cout << endl;
    getch();
}
```

Output -

enter base length of triangle : 2  
enter height length of triangle : 6

area = 0

### Program 5

//WAP to print the basic detail of student using diff.  
data type in C++.

```
#include<iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    char name[50];
    int rollno;
    float fee;
    char add[100];
    cout << "the name of student = ";
    cin >> name;
    cout << "the name is = " << name << endl;
    cout << "enter the roll no = ";
    cin >> rollno;
    cout << "roll no is = " << rollno << endl;
    cout << "enter the fee = ";
    cin >> fee;
    cout << "fee is = " << fee << endl;
    cout << "enter the add = ";
    cin >> add;
    cout << "add is = " << add << endl;
    getch();
}
```

## Output -

enter the name of student = abhinav

the name is = abhinav

enter the roll no = 362

roll no is = 362

enter the fee = 35,300

fee is = 35,300

enter the add = IP Bsn

add is = IP

## Program 6

//NAP to use Cin & Cout by using user-defined  
functions.

```
#include <iostream.h>
#include <conio.h>
void marvel()
{
    cout << "Avengers Endgame";
}
void main()
{
    clrscr();
    marvel(); //call
    getch();
}
```

Output -

Avengers Endgame

## Program 7

//WAP to use Enum data type in C++.

```
#include <iostream.h>
#include <conio.h>
enum week
{
    Sun,
    mon,
    Tue,
    Wed,
    thur,
    fri,
    Sat,
}
void main()
{
    clrscr();
    week day;
    day = Sun;
    for (day = Sun; day <= Sat; day++)
    {
        cout << "day is " << day << endl;
    }
    getch();
}
```

## Output -

day is 0  
day is 1  
day is 2  
day is 3  
day is 4  
day is 5  
day is 6

## Program 8

//WAP to use new/delete operator in C++

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int* value = new int;
    char* alphabet = new char;
    float* point = new float;
    double* jyaada = new double;

    *alphabet = 'A';
    *value = 23;
    *point = 12.5;
    *jyaada = 2345.856463;

    cout << *alphabet << endl;
    cout << *value << endl;
    cout << *point << endl;
    cout << *jyaada << endl;

    delete alphabet;
    delete value;
    delete point;
    delete jyaada;

    getch();
}
```

Output -

A

23

12.5

2345. 856463

Program 9

// WAP to use of structure in C++.

```

#include <iostream.h>
#include <conio.h>
struct books
{
    char title [50];
    float price;
    int pages;
    char author [50];
    char sub [30];
};

void main()
{
    clrscr();
    books s1; // instance
    cout << "enter title";
    cin >> s1.title;
    cout << "title is " << s1.title << endl;
    cout << "enter price";
    cin >> s1.price;
    cout << "price is " << s1.price << endl;
    cout << "enter pages";
    cin >> s1.pages;
    cout << " pages is " << s1.pages << endl;
    cout << "enter the name of author" << endl;
    cin >> s1.author;
    cout << "author is " << s1.author << endl;
    cout << "enter sub";
    cin >> s1.sub;
    cout << " sub is " << s1.sub << endl;
    getch();
}

```

## Output -

enter title hero

title is hero

enter price 230

price is 230

enter pages 40

pages is 40

enter the name of author sahani

author is sahani

enter sub War

Sub is War.

## PROGRAM - 10

//WAP to demonstrate use of reference variable.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int a = 20;
    int &Abhinav = a;
    cout << "value of a = " << Abhinav;
    getch();
}
```

Output -

value of a = 20

## Program 11

WAP to use scope resolution operator in C++

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int abc = 30; //Global variable
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int abc = 20;
```

```
cout << "local variable is = " << abc << endl;
```

```
cout << "global variable is = " << ::abc;
```

```
getch();
```

```
}
```

Output -

local variable is = 20

global variable is = 30

## Program 12

//WAP to accessing data members of class in C++

```
#include <iostream.h>
#include <conio.h>

class section
{
public:
    char name [50];
    int id;
    void input()
    {
        cout << "enter your name ";
        cin >> name;
        cout << "enter your id ";
        cin >> id;
        cout << name << endl;
        cout << id;
    }
};

void main()
{
    clrscr();
    section A;
    A.input();           //function call
    getch();
}
```

Output -

enter your name ABHINAV

enter your id 362

ABHINAV

362

## Program 13

//WAP to adding two no. using class in C++.

```
#include <iostream.h>
#include <conio.h>
class mathematics
{
public:
    int a, b;
    void input()
    {
        cout << "input two numbers \n";
        cin >> a >> b;
    }
    void add()
    {
        cout << "Result is: " << a+b;
    }
};

void main()
{
    clrscr();
    mathematics A;
    A.input();
    A.add();
    getch();
}
```

Output -

input two numbers

3

2

Result is : 5

## Program 14

//WAP to show the concepts of encapsulation using class in C++.

```
#include <iostream.h>
#include <conio.h>
class encapsulation
{
private:
    float radius;           // data member
    float area;
public:
    void circle()           // member function.
    {
        cout << "enter a radius = ";
        cin >> radius;
        area = 3.14 * radius * radius;
        cout << "area of circle = " << area;
    }
};

void main()
{
    clrscr();
    encapsulation box;
    box.circle();
    getch();
}
```

Output -

enter a radius

2

area of circle is = 12.56

## PROGRAM 15

//WAP to using multiple func. in class for mathematical operation.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class math
```

```
{
```

```
public:
```

```
int a,b,c;
```

```
void add()
```

```
{
```

```
cout << "enter first number = ";
```

```
cin >> a;
```

```
cout << "enter second number = ";
```

```
cin >> b;
```

```
c = a + b;
```

```
cout << "add of two numbers = " << c << endl;
```

```
}
```

```
void sub()
```

```
{
```

```
cout << "enter first number = ";
```

```
cin >> a;
```

```
cout << "enter second number = ";
```

```
cin >> b;
```

```
c = a - b;
```

```
cout << "sub of two numbers = " << c;
```

```
}
```

```
};
```

```
void main()
```

```
{
```

```
clrscr();
```

```
math ar;  
ar.add();  
ar.sub();  
getch();  
}
```

## Output -

enter first number = 2

enter second number = 3

add of two number = 5

enter first number = 2

enter second number = 3

Sub of two number = -1

Program 16

11WAP to show the concept of abstraction using  
math.h header file.

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
void main()
{
    clrscr();
    int n, power, result;
    cout << "enter a no: ";
    cin >> n;
    cout << " enter power for given no. ";
    cin >> power;
    result = pow(n, power);
    cout << " result " << result;
    getch();
}
```

Output -

enter a no: 2

enter power for given no. 2  
result 4

### PROGRAM 17

//WAP to show the concept of abstraction using math.h header file in class format.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
#include <math.h>
```

```
class math
```

```
{
```

```
public:
```

```
int n, power, result;
```

```
void square()
```

```
{
```

```
cout << "enter a no. ";
```

```
cin >> n;
```

```
cout << "enter power ";
```

```
cin >> power;
```

```
result = pow( n, power );
```

```
cout << "result " << result;
```

```
} ;
```

```
void main()
```

```
{
```

```
clrscr();
```

```
math Ar;
```

```
Ar.square();
```

```
getch();
```

```
}
```

Output -

enter a no. 3

enter power 3

result 27

**Output -**

enter your name = Abhinav  
enter your id 362  
Abhinav  
362

**Program 18**

//WAP to show the concept of local class in C++.

```
#include <iostream.h>
#include <conio.h>
void local()
{
    class student
    {
        public:
            int id;
            char name[50];
            void details();
    };
    cout << "enter your name";
    cin >> name;
    cout << " enter your id ";
    cin >> id;
    cout << name << endl << id;
}
student A;
A.details();
}

void main()
{
    clrscr();
    local();
    getch();
}
```

Output -

enter a value = 4  
value of a = 4

Program 19

|| WAP to show the concept of nested class use of resolution operator in C++

```
#include <iostream.h>
#include <conio.h>
class A
{
public:
    class B
    {
public:
        int a;
        void marvel()
        {
            cout << "enter a value= ";
            cin >> a;
            cout << "value of a = " << a;
        }
    };
};

void main()
{
    clrscr();
    A::B obj;           // resolution operator
    obj.marvel();
    getch();
}
```

Output -

enter two no. 2  
2  
22

Program 20

// WAP to print outside declaration of member function using class.

```
#include <iostream.h>
#include <conio.h>
class scalar
{
public:
    int a,b;
    void input();
    void marvel();
};

void scalar:: input()
{
    cout<<"enter two no. ";
    cin>>a>>b;
}

void scalar:: marvel()
{
    cout<<a<<b;
}

void main()
{
    clrscr();
    scalar S1;
    S1.input();
    S1.marvel();
    getch();
}
```

Output -

today's date is 27 7 2024

Program 21

1/WAP to initialization of abstraction in C++.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
#include <math.h>
```

```
class A
```

```
{
```

```
public:
```

```
int day;
```

```
int month;
```

```
int year;
```

```
void date (int day1, int month1, int year1)
```

```
{
```

```
day = day1;
```

```
month = month1;
```

```
year = year1;
```

```
}
```

```
void marvel () {
```

```
cout << day << " " << month << " " << year;
```

```
}
```

```
void main () {
```

```
clrscr();
```

```
A w;
```

```
cout << "today's date is ";
```

```
w.date (27, 07, 2024);
```

```
w.marvel();
```

```
getch();
```

```
}
```

Output -

cube is = 125  
multiplication is = 10

Program 22:

//WAP to show the concept of Inline function without class in C++.

```
#include <iostream.h>
#include <conio.h>
inline int hulk (int a)
{
    return (a*a*a);
}
inline double than (int a, int b)
{
    return (a*b);
}
void main()
{
    clrscr();
    cout << "cube is = " << hulk(5) << endl;
    cout << "multiplication = " <<
    than (2,5);
    getch();
}
```

### Output -

```
enter two nos 5 2  
the sum is: 7  
the sub is: 3  
the multiply is: 10  
the divide is: 2
```

### Program 23

```
//WAP to show the concept of Inline function  
with the class in C++.  
#include <iostream.h>  
#include <conio.h>  
class all {  
private:  
    int a, b, sum, sub, mul, div;  
public:  
    void getdata();  
    void add();  
    void subtract();  
    void multiply();  
    void division();  
};  
inline void all :: getdata () {  
    cout << "enter two no.s ";  
    cin >> a >> b; }  
inline void all :: add () {  
    sum = a + b;  
    cout << "the sum is: " << sum << endl; }  
inline void all :: subtract () {  
    sub = a - b;  
    cout << "the sub is: " << sub << endl; }  
inline void all :: multiply () {  
    mul = a * b;  
    cout << "the multiply is: " << mul << endl; }  
inline void all :: division () {  
    div = a / b;  
    cout << "the divide is: " << div << endl; }  
void main ()  
{ }
```

```
clscur();  
all ab;  
ab.getData();  
ab.add();  
ab.subtract();  
ab.multiply();  
ab.division();  
getch();
```

}

Output -

3  
7  
3  
3

### Program 25

//WAP to demonstrate the concept of static data member and member function in C++.

```
#include <iostream.h>
#include <conio.h>
class open
{
    int a,b;
    static int c;
public:
    void input (int n, int y)
    {
        a = n;
        b = y;
        c = c + 1;
    }
    static void print ()
    {
        cout << c;
    }
    void show ()
    {
        cout << a << endl;
        cout << b << endl;
        cout << c << endl;
    }
};

int open :: c;
void main ()
{
    clrscr ();
    open aa, bb, cc;
    aa.input (3, 7);
    bb.input (9, 11);
    cc.input (10, 12);
    aa.show ();
    open :: print ();
    getch ();
}
```

Output -  
enter a no. 8  
0  
1  
1  
2  
3  
5  
8  
13

### Program 24

//WAP to Print Fibonacci series in C++.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int n, a=0, b=1, c;
    cout<<" enter a no. ";
    cin>>n;
    for( int i = 1 ; i <=n ; i++)
    {
        cout<<a<<endl;
        c = a+b;
        a= b;
        b = c;
    }
    getch();
}
```

Output -

enter the name Abhinav  
Abhinav

Program 26

//WAP to print Inside declaration of Constructor  
using class.

```
#include <iostream.h>
#include <conio.h>
class bca
{
private:
    char name [50];
public:
    bca()
    {
        cout << " enter the name ";
        cin >> name;
    }
    void display()
    {
        cout << name;
    }
};
void main()
{
    bca();
    bca Sem;
    Sem.display();
    getch();
}
```

Output -

enter id 362  
362

Program 27

// WAP to print outside declaration of constructor using class.

#include <iostream.h>

#include <conio.h>

class world

{

private:

int id;

public:

world();

void display(); } ;

world:: world() // scope resolution

{

cout << " enter id";

cin >> id;

}

void world:: display()

{

cout << id;

}

void main()

{

clrscr();

world s1;

s1. display();

getch();

}

Program 28

//WAP to show the concept of default Constructor

```
#include <iostream.h>
#include <conio.h>
class ghar
{
private:
    double length;
public:
    ghar()           // default constructor
    {
        length = 5.8;
        cout << "my wall is created" << endl;
        cout << " my wall length = " << length;
    }
    void main()
    {
        clrscr();
        ghar room;      //invoked constructor
        getch();
    }
}
```

Output -

my wall is created  
my wall length = 5.8

### Program 29

//WAP to show the concept of parameterised constructor

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class ghar :
```

```
{
```

```
private :
```

```
double length, breadth, area;
```

```
public :
```

```
ghar(double l, double b) :
```

```
length = l;
```

```
breadth = b;
```

```
}
```

```
void calculate ()
```

```
{
```

```
area = length * breadth;
```

```
cout << "my wall area is = " << area;
```

```
}
```

```
}
```

```
void main ()
```

```
{
```

```
clrscr();
```

```
ghar w(7.9, 8.6); //initializing of object
```

```
w.calculate();
```

```
} getch();
```

```
}
```

Output =  
my wall area is = 67.94

## Program 30

//WAP to show the concept of copy Constructor.

```
#include<iostream.h>
#include<conio.h>
class demo
{
private:
    int a,b;
public:
    demo(int x, int y)
    {
        a = x;
        b = y;
    }
    demo (demo & parte)
    {
        a = parte.a;
        b = parte.b;
    }
    void show()
    {
        cout << "value of a = " << a << endl;
        cout << "value of b = " << b << endl;
    }
};

void main ()
{
    clrscr();
    demo aa(7,8);
    aa.show();
    demo bb = aa;
    getch();
}
```

Output :-

value of a = 7  
value of b = 8

## Program 31

//WAP to show the concept of constructor overloading.

```
#include <iostream.h>
```

```
# include <conio.h>
```

```
class size
```

```
{
```

```
private:
```

```
int area;
```

```
public:
```

```
size()
```

```
{
```

```
area = 0;
```

```
}
```

```
size(int side)
```

```
{
```

```
area = side * side;
```

```
}
```

```
size( int l, int b )
```

```
{
```

```
area = l * b;
```

```
}
```

```
size( float radius )
```

```
{
```

```
area = 4 * 3.14 * radius * radius;
```

```
}
```

```
void show()
```

```
{
```

```
cout << area << endl;
```

```
} ;
```

```
void main()
```

```
{
```

```
char s[10];  
cout << "my initialize value of area is = " << endl;  
size aa;  
aa.show();  
cout << "area of square = " << endl;  
size bb(5);  
bb.show();  
cout << "area of a length = " << endl;  
size cc(5,9);  
cc.show();  
cout << "area of a radius = " << endl;  
size dd(2,5);  
dd.show();  
getch();  
3
```

Output -

my initialize value of area is =  
0

area of square =  
25

area of a length =  
45

area of a radius =  
10

## Program 32

Page No.	
Date	

//WAP to show the concept of Dynamic Constructors.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Dynamic {
```

```
private:
```

```
int *p;
```

```
int size;
```

```
public:
```

```
Dynamic (int s);
```

```
void input();
```

```
void output();
```

```
};
```

```
Dynamic :: Dynamic (int s) {
```

```
size = s;
```

```
p = new int [size];
```

```
void Dynamic :: input() {
```

```
cout << "enter values " << endl;
```

```
for (int i = 0; i < size; i++) {
```

```
cin >> p[i];
```

```
}
```

```
void Dynamic :: output() {
```

```
cout << "my entered values are " << endl;
```

```
for (int i = 0; i < size; i++) {
```

```
cout << p[i];
```

```
}
```

```
void main() {
```

```
clrscr();
```

```
int n;
```

```
cout << "enter the size " << endl;
```

```
cin >> n;
```

```
Dynamic aa (n);
```

```
aa.input();
```

```
aa.output();
```

```
} getch();
```

Output -

enter the size

4

enter values

5

6

6

7

my entered value are  
5 6 6 7

## Program 33

Page No.		
Date		

//WAP to show the concept of Destructor in C++

```
#include <iostream.h>
#include <conio.h>
class Abc
{
public:
    Abc()           // constructor
    {
        cout << "constructor is invoked" << endl;
    }
    ~Abc()          // destructor
    {
        cout << "destructor is invoked" << endl;
    }
    void main()
    {
        clrscr();
    }
    Abc aa;
}
 getch();
```

Output =

Constructor is invoked  
destructor is invoked

## Program 34

Page No.		
Date	/	/

//WAP to show the concept of Default Parameter/argument.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class op
```

```
{
```

```
public:
```

```
int sum(int x, int y, int z = 0)
```

```
{
```

```
    return (x+y+z);
```

```
}
```

```
void main()
```

```
{
```

```
clsur();
```

```
op obj;
```

```
cout << "sum = " << obj.sum(3,4) << endl;
```

```
cout << "next sum = " << obj.sum(4,4,4);
```

```
getch();
```

```
3
```

Output -

Sum = 7

next sum = 12