

Unit 1 medium

Find the Square root of a number in C++ using IF-ELSE statement?
--

1296

36

```
#include <iostream>
```

```
#include <cmath>
```

```
Using namespace std
```

```
int main() {
```

```
    int number;
```

```
    std::cout << "Enter a number: ";
```

```
    std::cin >> number;
```

```
    if (number >= 0) {
```

```
        double squareRoot = sqrt(number);
```

```
        cout << "Square root of " << number << " is: " << squareRoot << endl;
```

```
    } else {
```

```
        std::cout << "Square root of a negative number is imaginary and cannot be calculated."
<< std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

2.

Find the Cube root of a number in in C++ using any conditional statements?
--

9

3

```
#include <iostream>
```

```
#include <cmath>
```

```

int main() {
    int number;

    std::cout << "Enter a number: ";

    std::cin >> number;

    double cubeRoot = cbrt(number);

    std::cout << "Cube root of " << number << " is: " << cubeRoot << std::endl;

    return 0;
}

```

3.

Write a C++ program to check if a number is perfect or not using IF condition?
--

6

6 is a perfect number

```

#include <iostream>

using namespace std;

int main() {
    int number, sum = 0;

    cout << "Enter a number: ";

    cin >> number;

    for (int i = 1; i < number; ++i) {
        if (number % i == 0) {
            sum += i;
        }
    }

    if (sum == number) {
        cout << number << " is a perfect number" << endl;
    } else {
        cout << number << " is not a perfect number" << endl;
    }
}

```

```

    }
return 0;
}

```

4.

Write a C++ program to find the smallest element missing in a sorted array using WHILE loop?
--

{0, 1, 2, 3, 5, 6, 7}

4

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int arr[] = {0, 1, 2, 3, 5, 6, 7};
```

```
    int size = sizeof(arr) / sizeof(arr[0]);
```

```
    int left = 0;
```

```
    int right = size - 1;
```

```
    if (arr[0] != 0) {
```

```
        cout << "The smallest missing element in the sorted array is: 0" << endl;
```

```
        return 0;
```

```
    }
```

```
    while (left <= right) {
```

```
        if (arr[left] == left) {
```

```
            left++;
```

```
        } else {
```

```
            cout << "The smallest missing element in the sorted array is: " << left << endl;
```

```
            return 0;
```

```
        }
```

```
    }
```

```
    cout << "The smallest missing element in the sorted array is: " << arr[size - 1] + 1 << endl;
```

```
    return 0;
}
```

5.

Write a program in C++ to find the sum of n natural numbers using FOR loop?

Enter the value of n : 3

Enter the numbers : 2 5 8

```
#include <iostream>

using namespace std;

int main()
{
    int n;

    cout << "Enter the value of n: ";

    cin >> n;

    int sum = 0;

    int num;

    cout << "Enter the numbers: ";

    for (int i = 0; i < n; ++i)
    {
        cin >> num;

        sum += num;
    }

    cout << "The sum of " << n << " natural numbers is: " << sum << endl;

    return 0;
}
```

6.above program using while loop.

```
#include <iostream>
```

```
using namespace std;
```

```

int main()
{
    int n;

    cout << "Enter the value of n: ";

    cin >> n;

    int sum = 0;

    int count = 0;

    int num;

    cout << "Enter the numbers: ";

    while (count < n)
    {
        cin >> num;

        sum += num;

        count++;
    }

    cout << "The sum of " << n << " natural numbers is: " << sum << endl;

    return 0;
}

```

7.above program using do while loop

```

#include <iostream>

using namespace std;

int main()
{
    int n;

    cout << "Enter the value of n: ";

    cin >> n;

    int sum = 0;

```

```

int count = 0;

int num;

cout << "Enter the numbers: ";

do
    {
        cin >> num;

        sum += num;

        count++;
    } while (count < n);

cout << "The sum of " << n << " natural numbers is: " << sum << endl;

return 0;
}

```

9. perfect or not using while loop.

```

#include <iostream>

using namespace std;

int main() {

    int number, sum = 0, remainder = 1;

    cout << "Enter a number: ";

    cin >> number;

    while (remainder < number) {

        if (number % remainder == 0) {

            sum += remainder;

        }

        remainder++;

    }

    if (sum == number) {

```

```

        cout << number << " is a perfect number" << endl;
    } else {
        cout << number << " is not a perfect number" << endl;
    }
    return 0;
}

```

Unit 2

Develop a c++ program for default arguments
Enter the value: 10 15 25 30
80

```
#include <iostream>
```

```
using namespace std;
```

```
// Function with default arguments
```

```
int sum(int a = 0, int b = 0, int c = 0, int d = 0) {
```

```
    return a + b + c + d;
```

```
}
```

```
int main() {
```

```
    int num1, num2, num3, num4;
```

```
    cout << "Enter the value: ";
```

```
    cin >> num1 >> num2 >> num3 >> num4;
```

```
    int result = sum(num1, num2, num3, num4);
```

```
    cout << "Sum is: " << result << endl;
```

```
    return 0;
```

```
}
```

2.

Develop a c++ program for adding the number using function overloading concept
--

Enter the value for a, b, c : 10 20 60
The value of addition using two parameter is 30
The value of addition using three parameter is 60

```
#include <iostream>

using namespace std;

// Function to add two integers
int add(int a, int b) {
    return a + b;
}

// Function overloading to add three integers
int add(int a, int b, int c) {
    return a + b + c;
}

int main() {
    int num1, num2, num3;

    cout << "Enter the value for a, b, c: ";

    cin >> num1 >> num2 >> num3;

    // Call the appropriate function based on the number of parameters entered
    int sum_two = add(num1, num2);
    int sum_three = add(num1, num2, num3);

    cout << "The value of addition using two parameters is: " << sum_two << endl;
    cout << "The value of addition using three parameters is: " << sum_three << endl;

    return 0;
}

3.
```

Declare a class box, with length(Public variable) and width(Private variable) use set width ()and get width() function to set the width and print the length and width ..
Enter the Length of box :6
Enter the Width of box :9

Length of box : 6 Width of box : 9

```
#include <iostream>

using namespace std;

class Box
{
private:
    int width;
public:
    int length;
    void setWidth(int w)
    {
        width = w;
    }
    int getWidth()
    {
        return width;
    }
};

int main()
{
    Box boxObj;

    cout << "Enter the Length of box: ";
    cin >> boxObj.length;

    cout << "Enter the Width of box: ";
    int width;
    cin >> width;
```

```

boxObj.setWidth(width);

cout << "Length of box: " << boxObj.length << endl;

cout << "Width of box: " << boxObj.getWidth() << endl;

return 0;

}

4.

```

Develop a c++ program for matrix multiplication using arrays
enter the number of row=3 enter the number of column=3 enter the first matrix element= 1 2 3 1 2 3 1 2 3 enter the second matrix element= 1 1 1 2 1 2 3 2 1
multiply of the matrix= 14 9 8 14 9 8 14 9 8

```

#include <iostream>

using namespace std;

int main()

{

    int row1, col1, row2, col2;

    cout << "Enter the number of rows and columns for the first matrix: ";

    cin >> row1 >> col1;

    int matrix1[row1][col1];

    cout << "Enter elements for the first matrix:" << endl;

    for (int i = 0; i < row1; ++i)

        for (int j = 0; j < col1; ++j)

```

```

        cin >> matrix1[i][j];

    cout << "Enter the number of rows and columns for the second matrix: ";

    cin >> row2 >> col2;

    if (col1 != row2) {

        cout << "Matrix multiplication not possible! Column of the first matrix should be equal
to row of the second matrix." << endl;

        return 0;

    }

    int matrix2[row2][col2];

    cout << "Enter elements for the second matrix:" << endl;

    for (int i = 0; i < row2; ++i)

        for (int j = 0; j < col2; ++j)

            cin >> matrix2[i][j];

    int result[row1][col2] = {0};

    for (int i = 0; i < row1; ++i)

        for (int j = 0; j < col2; ++j)

            for (int k = 0; k < col1; ++k)

                result[i][j] += matrix1[i][k] * matrix2[k][j];

    cout << "Resultant matrix (Multiplication):" << endl;

    for (int i = 0; i < row1; ++i) {

        for (int j = 0; j < col2; ++j)

            cout << result[i][j] << " ";

        cout << endl;

    }

    return 0;

}

5. #include <iostream>

```

```
using namespace std;

class Car
{
private:
    int id;

    string name;

    int marks;

    static int count;

public:
    Car(int id, string name, int marks) {
        this->id = id;

        this->name = name;

        this->marks = marks;

        count++;
    }

    void display() {
        cout << "Id of the Car: " << id << endl;

        cout << "Name of the Car: " << name << endl;

        cout << "Marks: " << marks << endl << endl;
    }

    static void displayCount() {
        cout << "No. of objects created in the class: " << count << endl;
    }
};

int Car::count = 0;

int main() {
    int id, marks;
```

```

    string name;

    Car car1(101, "Ferrari", 10);

    Car car2(205, "Mercedes", 9);

    car1.display();

    car2.display();

    Car::displayCount();

    return 0;

}

6.

```

Develop a C++ program to perform different arithmetic operations such as addition, subtraction, division, modulus and multiplication switch case

Calculator: 1.Addition 2. Subtraction. 3.Multiplication 4. Division 5.Modulus Enter your choice:1 Number 1: 20 Number 2:30
--

The value of addition is 50

```

#include <iostream>

using namespace std;

int main()
{
    int choice;

    int num1, num2, result;

    cout << "1. Addition\n";

    cout << "2. Subtraction\n";

    cout << "3. Multiplication\n";

    cout << "4. Division\n";

```

```
cout << "5. Modulus\n";

cout << "Enter your choice: ";

cin >> choice;

cout << "Number 1: ";

cin >> num1;

cout << "Number 2: ";

cin >> num2;

switch (choice)

{

case 1:

    result = num1 + num2;

    cout << "The value of addition is " << result << endl;

    break;

case 2:

    result = num1 - num2;

    cout << "The value of subtraction is " << result << endl;

    break;

case 3:

    result = num1 * num2;

    cout << "The value of multiplication is " << result << endl;

    break;

case 4:

    if (num2 != 0) {

        result = num1 / num2;

        cout << "The value of division is " << result << endl;

    } else {

        cout << "Cannot divide by zero!" << endl;
```

```

    }

    break;

case 5:

    result = num1 % num2;

    cout << "The value of modulus is " << result << endl;

    break;

default:

    cout << "Invalid choice!" << endl;

    break;

}

return 0;

}

7.

```

Develop a Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

Enter Name of the Employee : Suresh
Enter Address of the Employee : Vetri Nagar
Enter ID of the Employee :7001
Enter Mobile Number : 9898989898
Enter E-Mail ID of the Employee : aff@gmail.com
ENTER THE BASIC PAY OF THE PROGRAMMER => 80000

```

=====
PROGRAMMER PAYMENT SLIP
=====
BASIC PAY => 80000
DEARNESS ALLOWANCE => 77600
HOUSE RENT ALLOWENCE => 8000
PROVIDENT FUND => 9600
CLUB FUND => 800
GROSS PAY => 175200
NET PAY => 164800

```

```
#include <iostream>
```

```
#include <string>

using namespace std;

class Employee {
public:
    string emp_name;
    int emp_id;
    string address;
    string mail_id;
    long long mobile_no;
    void getEmployeeDetails()
    {
        cout << "Enter Name of the Employee: ";
        cin >> emp_name;
        cout << "Enter Address of the Employee: ";
        cin>> address;
        cout << "Enter ID of the Employee: ";
        cin >> emp_id;
        cout << "Enter Mobile Number: ";
        cin >> mobile_no;
        cout << "Enter E-Mail ID of the Employee: ";
        cin >> mail_id;
    }
};

class Programmer : public Employee
{
public:
    double basic_pay;
```



```

double da;

double hra;

double pf;

double club_fund;

void getBasicPay()
{
    cout << "ENTER THE BASIC PAY OF THE PROGRAMMER => ";

    cin >> basic_pay;
}

void calculateSalary()
{
    da = 0.97 * basic_pay;

    hra = 0.1 * basic_pay;

    pf = 0.12 * basic_pay;

    club_fund = 0.001 * basic_pay;

    double gross_pay = basic_pay + da + hra;

    double net_pay = gross_pay - (pf + club_fund);

    cout << "=====\n";

    cout << "PROGRAMMER PAYMENT SLIP\n";

    cout << "=====\n";

    cout << "BASIC PAY => " << basic_pay << endl;

    cout << "DEARNESS ALLOWANCE => " << da << endl;

    cout << "HOUSE RENT ALLOWANCE => " << hra << endl;

    cout << "PROVIDENT FUND => " << pf << endl;

    cout << "CLUB FUND => " << club_fund << endl;

    cout << "GROSS PAY => " << gross_pay << endl;

    cout << "NET PAY => " << net_pay << endl;
}

```

```

    }
};

int main() {
    Programmer programmer;

    programmer.getEmployeeDetails();

    programmer.getBasicPay();

    programmer.calculateSalary();

    return 0;
}

8.

```

Write a c++ program to remove duplicates from the sorted array	
Sample Input:	
Array = {15, 14, 25, 14, 32, 14, 31}	
Sample Output:	
Sorted Array = {14, 15, 25, 31, 32}	
1.	{16, 16, 16 16, 16}
2.	{0, 0, 0, 0}
3.	{-12, -78, -35, -42}
4.	{1,2,3,7,8,9,4,5,6}
5.	{1-2,2-3,3-4,4-5,5-6}

```

#include<iostream>

using namespace std;

int main ()
{
    int A[10], B[10], n, i, j, k = 0;

    cout << "Enter size of array : ";

    cin >> n;

    cout << "Enter elements of array : ";

    for (i = 0; i < n; i++)
        cin >> A[i];

```

```

for (i = 0; i < n; i++)
{
    for (j = 0; j < k; j++)
    {
        if (A[i] == B[j])
            break;
    }
    if (j == k)
    {
        B[k] = A[i];
        k++;
    }
}

cout << "Repeated elements after deletion : ";

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

    cout << B[i] << " ";

return 0;
}

```

9. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is $60 \geq$ and < 75 , then the grade is First Division. If aggregate is $50 \geq$ and < 60 , then the grade is Second Division. If aggregate is $40 \geq$ and < 50 , then the grade is Third Division. Else the grade is Fail.

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int marks[4],total=0,aggregate;
```

```
    cout<<"enter the marks of the student";  
    for(int i=0;i<4;i++)  
    {  
        cin>>marks[i];  
        total+=marks[i];  
    }  
    aggregade=total/4.0;  
    cout<<total<<endl;  
    cout<<aggregade<<endl;  
    if(aggregade>=75)  
    {  
        cout<<"distinction";  
    }  
    else if(aggregade>=60)  
    {  
        cout<<"first division";  
    }  
    else if(aggregade>=50)  
    {  
        cout<<"second division";  
    }  
    else  
    {  
        cout<<"fail";  
    }  
    return 0;  
}
```

10.

Develop a largest class with a,b, and m as member. Use setdata () for setting the data and friend void find_Max (largest) function for finding the largest number.

Enter the first no: 52

Enter the second no: 63

Maximum no is 63

```
#include <iostream>

using namespace std;

class largest {
private:
    int a, b;
public:
    void setdata(int x, int y) {
        a = x;
        b = y;
    }

    friend void find_Max(largest obj);
};

void find_Max(largest obj) {
    int max = (obj.a > obj.b) ? obj.a : obj.b;
    cout << "The largest number is: " << max << endl;
}

int main() {
    largest obj;
    int x, y;
    cout << "Enter two numbers: ";
    cin >> x >> y;
    obj.setdata(x, y);
```

```

    find_Max(obj);

    return 0;

}

```

Unit 3;

1. Write a program to find area of Circle, Rectangle and Triangle using constructor overloading.
--

3

6,7

2,3

28.26

42

3

```

#include<iostream>

```

```

#include<cmath>

```

```

using namespace std;

```

```

class area

```

```

{

```

```

    public:

```

```

        double radius;

```

```

        int length;

```

```

        int breadth;

```

```

        int base;

```

```

        int height;

```

```

        area()

```

```

        {

```

```

            cin>>radius;

```

```

            cin>>length;

```

```

            cin>>breadth;

```

```

            cin>>base;

```

```

        cin>>height;
    }
    ~area()
    {
        cout<<"object is destroyed";
    }
    void areadetails()
    {
        cout<<3.14*radius*radius<<endl;
        cout<<length*breadth<<endl;
        cout<<0.5*base*height<<endl;
    }
};

int main()
{
    area a;
    a.areadetails();
}

```

2. Write a program to find Cube, Cylinder using constructor overloading

```

#include<iostream>

using namespace std;

class volume
{
    public:
        int side;
        double radius;
        double height;

```

```

        volume()
        {
            cin>>side;

            cin>>radius;

            cin>>height;

        }

~volume()
{
    cout<<"object is destroyed";
}

void volumedetails()
{
    cout<<"volume of cube is"<<side*side*side<<endl;

    cout<<"volume of cylinder is"<<3.14*radius*radius*height<<endl;

}

};

int main()
{
    volume v;

    v.volumedetails();

}

```

3. Write a program to declare the constructor inside the class, and then define it outside of the class by specifying the name of the class.

```
#include <iostream>
```

```
class MyClass {
```

```
private:
```



```
int myValue;
```

```
public:
```

```
    MyClass(int value);
```

```
    void displayValue();
```

```
};
```

```
MyClass::MyClass(int value) : myValue(value) {}
```

```
void MyClass::displayValue() {
```

```
    std::cout << "Value: " << myValue << std::endl;
```

```
}
```

```
int main() {
```

```
    MyClass obj(42);
```

```
    obj.displayValue();
```

```
    return 0;
```

```
}
```

4. Write a program in C++ to print Floyd's Triangle by using the constructor destructor

```
#include <iostream>
```

```
class FloydTriangle
```

```
{
```

```
public:
```

```
    FloydTriangle(int rows)
```

```
    {
```

```
        int number = 1;
```

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

```

        {
            for (int j = 1; j <= i; ++j)
            {
                std::cout << number << " ";
                ++number;
            }
            std::cout << std::endl;
        }
    }
    ~FloydTriangle() {
        std::cout << "\nDestructor Called";
    }
};

int main()
{
    int numRows;

    std::cout << "Enter the number of rows for Floyd's Triangle: ";

    std::cin >> numRows;

    FloydTriangle triangle(numRows);

    return 0;
}

```

5. Write a program in C++ to convert a decimal number into binary without using an array by using the constructor overloading.

```

#include <iostream>

class DecimalToBinary
{
public:

```

```

int decimalNumber;

long long binaryNumber;

DecimalToBinary(int num) : decimalNumber(num), binaryNumber(0)
{
    convertToBinary();
}

void convertToBinary()
{
    long long tempBinary = 0, multiplier = 1;
    while (decimalNumber > 0)
    {
        tempBinary += (decimalNumber % 2) * multiplier;
        decimalNumber /= 2;
        multiplier *= 10;
    }
    binaryNumber = tempBinary;
}

void displayBinary() const
{
    std::cout << "Binary equivalent: " << binaryNumber << std::endl;
}

};

int main()
{
    int decimalNum;

    std::cout << "Enter a decimal number: ";

    std::cin >> decimalNum;

```

```
DecimalToBinary converter(decimalNum);  
converter.displayBinary();  
return 0;  
}
```