1.implicit

#include<iostream>

using namespace std;

int main()

{

int a=10;

char b='a';

a=b+a;

float c=a+1.0;

cout<<"a:"<<a<<"\nb:"<<b<<"\nc:"<<c;

}

Output:

a:107

b:a

c:108

2.explicit

#include<iostream>

using namespace std;

int main()

{

double x=1.574;

int add =(int)x+1;

cout<<"add:"<<add;

float y=3.5;

int val=static\_cast<int>(y);

cout<<"\nvalue:"<<val;

}

Output:

add:2

value:3

3.String upper case to lower case

#include <iostream>

#include <string>

using namespace std;

class StringConverter {

public:

// Constructor

StringConverter() {}

// Destructor

~StringConverter() {}

// Function to convert a string to upper case

string toUpperCase(string str) {

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

str[i] = toupper(str[i]);

}

return str;

}

// Function to convert a string to lower case

string toLowerCase(string str) {

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

str[i] = tolower(str[i]);

}

return str;

}

};

int main() {

// Create an object of the StringConverter class

StringConverter converter;

// Get the input string from the user

cout << "Enter a string: ";

string str;

cin >> str;

string upperCaseString = converter.toUpperCase(str);

string lowerCaseString = converter.toLowerCase(str);

cout << "The upper case string is: " << upperCaseString << endl;

cout << "The lower case string is: " << lowerCaseString << endl;

return 0;

}

Output:

Enter a string: UPPERCASE

The upper case string is: UPPERCASE

The lower case string is: uppercase

4.Odd or even

#include<iostream>

int main() {

int number;

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

std::cin >> number;

if (number % 2 == 0) {

std::cout << "The number is Even" << std::endl;

} else {

std::cout << "The number is Odd" << std::endl;

}

return 0;

}

Output:

Enter a number: 24

The number is Even

5.Prime or Not

#include<iostream>

int main() {

int number;

// Get input from the user

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

std::cin >> number;

// Check if the number is less than 2 (not a prime number)

if (number < 2) {

std::cout << number << " is not prime." << std::endl;

}

else {

bool isPrime = true;

// Check for factors from 2 to the square root of the number

for (int i = 2; i \* i <= number; ++i) {

if (number % i == 0) {

isPrime = false;

break;

}

}

// Output the result based on the isPrime flag

if (isPrime) {

std::cout << number << " is prime." << std::endl;

} else {

std::cout << number << " is not prime." << std::endl;

}

}

return 0;

}

Output:

Enter a number: 21

21 is not prime.

6. Temperature in Celcius to Fahrenheit

#include<iostream>

int main() {

double celsius;

// Get input from the user

std::cout << "Enter temperature in Celsius: ";

std::cin >> celsius;

// Check if the input is valid

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter a valid number." << std::endl;

return 1; // Return an error code

}

// Convert Celsius to Fahrenheit

double fahrenheit = (celsius \* 9 / 5) + 32;

// Display the result

std::cout << "Temperature in Fahrenheit: " << fahrenheit << std::endl;

return 0;

}

Output:

Enter temperature in Celsius: 25

Temperature in Fahrenheit: 77

7.Palindrome or not

#include<iostream>

#include<string>

#include<cctype> // for std::toupper

bool isPalindrome(const std::string& str) {

// Iterate through the string and check for palindrome

int i = 0;

int j = str.length() - 1;

while (i < j) {

if (std::toupper(str[i]) != std::toupper(str[j])) {

return false;

}

i++;

j--;

}

return true;

}

int main() {

std::string input;

// Get input from the user

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

std::getline(std::cin, input);

// Check if the input is palindrome

if (isPalindrome(input)) {

std::cout << "It is palindrome." << std::endl;

} else {

std::cout << "It is not palindrome." << std::endl;

}

return 0;

}

Output:

Enter a number or string: 121

It is palindrome.

8. Square of a number

#include<iostream>

#include<cmath>

int main() {

double number;

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

std::cin >> number;

if (std::cin.fail() || number < 0) {

std::cerr << "Invalid input. Please enter a non-negative number." << std::endl;

return 1;

}

double squareRoot = sqrt(number);

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

return 0;

}

Output:

Enter a number: 12

Square root of 12 is: 3.4641

9.Cube root of a number

#include<iostream>

#include<cmath>

int main() {

double number;

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

std::cin >> number;

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter a valid number." << std::endl;

return 1;

}

double cubeRoot = cbrt(number);

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

return 0;

}

Output:

Enter a number: 5

Cube root of 5 is: 1.70998

10.Perfect number or not

#include<iostream>

int main() {

int number;

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

std::cin >> number;

if (std::cin.fail() || number < 1) {

std::cerr << "Invalid input. Please enter a positive integer." << std::endl;

return 1;

}

int sum = 0;

for (int i = 1; i <= number / 2; ++i) {

if (number % i == 0) {

sum += i;

}

}

if (sum == number) {

std::cout << number << " is a perfect number." << std::endl;

} else {

std::cout << number << " is not a perfect number." << std::endl;

}

return 0;

}

Output:

Enter a number: 24

24 is not a perfect number.

11.Perfect number or not using if else

#include<iostream>

int main() {

int number, sum = 0;

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

std::cin >> number;

if (std::cin.fail() || number < 1) {

std::cerr << "Invalid input. Please enter a positive integer." << std::endl;

return 1;

}

for (int i = 1; i < number; ++i) {

if (number % i == 0) {

sum += i;

}

}

if (sum == number) {

std::cout << number << " is a perfect number." << std::endl;

} else {

std::cout << number << " is not a perfect number." << std::endl;

}

return 0;

}

Output:

Enter a number: 12

12 is not a perfect number.

12. Smallest number missing in the array

#include <iostream>

int findSmallestMissingElement(int arr[], int size) {

int i = 0;

while (i < size) {

if (arr[i] == i + 1) {

++i;

} else if (arr[i] <= 0 || arr[i] > size || arr[arr[i] - 1] == arr[i]) {

++i;

} else {

std::swap(arr[i], arr[arr[i] - 1]);

}

}

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

if (arr[i] != i + 1) {

return i + 1;

}

}

return size + 1;

}

int main() {

int size;

std::cout << "Enter the size of the array: ";

std::cin >> size;

int arr[size];

std::cout << "Enter the sorted array elements: ";

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

std::cin >> arr[i];

}

int smallestMissing = findSmallestMissingElement(arr, size);

std::cout << "The smallest missing number is: " << smallestMissing << std::endl;

return 0;

}

Output:

Enter the size of the array: 5

Enter the sorted array elements: 4

7

5

8

9

The smallest missing number is: 1

13.Sum of n natural numbers using for loop.

#include<iostream>

int main() {

int n;

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

std::cin >> n;

if (std::cin.fail() || n < 0) {

std::cerr << "Invalid input. Please enter a non-negative integer for n." << std::endl;

return 1;

}

int sum = 0;

std::cout << "Enter " << n << " numbers: ";

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

int num;

std::cin >> num;

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter valid integers for numbers." << std::endl;

return 1;

}

sum += num;

}

std::cout << "Sum: " << sum << std::endl;

return 0;

}

Output:  
Enter the value of n: 5

Enter 5 numbers: 4

8

6

9

4

Sum: 31

14.Sum of n natural numbers using while loop.

#include<iostream>

int main() {

int n;

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

std::cin >> n;

if (std::cin.fail() || n < 0) {

std::cerr << "Invalid input. Please enter a non-negative integer for n." << std::endl;

return 1;

}

int sum = 0;

int i = 0;

std::cout << "Enter " << n << " numbers: ";

while (i < n) {

int num;

std::cin >> num;

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter valid integers for numbers." << std::endl;

return 1;

}

sum += num;

i++;

}

std::cout << "Sum: " << sum << std::endl;

return 0;

}

Output:

Enter the value of n: 5

Enter 5 numbers: 4

8

6

9

4

Sum: 31

15.Sum of n natural numbers using do while

#include<iostream>

int main() {

int n;

// Get the value of n from the user

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

std::cin >> n;

// Check if the input is valid

if (std::cin.fail() || n < 0) {

std::cerr << "Invalid input. Please enter a non-negative integer for n." << std::endl;

return 1; // Return an error code

}

int sum = 0;

int i = 1; // Start counting from 1 for natural numbers

// Get numbers from the user and calculate the sum using a DO-WHILE loop

std::cout << "Enter " << n << " numbers: ";

do {

int num;

std::cin >> num;

// Check if the input is valid

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter valid integers for numbers." << std::endl;

return 1; // Return an error code

}

sum += num;

i++;

} while (i <= n);

// Display the result

std::cout << "Sum: " << sum << std::endl;

return 0;

}

Output:

Enter the value of n: 5

Enter 5 numbers: 4

8

6

9

4

Sum: 31

16.Perfect number or not using for loop

#include<iostream>

int main() {

int num;

// Get the input number from the user

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

std::cin >> num;

// Check if the input is valid

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter a valid integer." << std::endl;

return 1; // Return an error code

}

if (num < 1) {

std::cerr << "Invalid input. Please enter a positive integer." << std::endl;

return 1; // Return an error code

}

int sum = 0;

// Check for perfect number using a FOR loop

for (int i = 1; i <= num / 2; ++i) {

if (num % i == 0) {

sum += i;

}

}

// Check if the sum of divisors is equal to the original number

if (sum == num) {

std::cout << num << " is a perfect number." << std::endl;

} else {

std::cout << num << " is not a perfect number." << std::endl;

}

return 0;

}

Output:

Enter a number: 121

121 is not a perfect number.

17. Perfect number or not using do while.

#include<iostream>

int main() {

int num;

// Get the input number from the user

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

std::cin >> num;

// Check if the input is valid

if (std::cin.fail()) {

std::cerr << "Invalid input. Please enter a valid integer." << std::endl;

return 1; // Return an error code

}

if (num < 1) {

std::cerr << "Invalid input. Please enter a positive integer." << std::endl;

return 1; // Return an error code

}

int sum = 0;

int i = 1; // Start with the first divisor

// Check for perfect number using a WHILE loop

while (i <= num / 2) {

if (num % i == 0) {

sum += i;

}

i++;

}

// Check if the sum of divisors is equal to the original number

if (sum == num) {

std::cout << num << " is a perfect number." << std::endl;

} else {

std::cout << num << " is not a perfect number." << std::endl;

}

return 0;

}

Output:

Enter a number: 121

121 is not a perfect number.

18.Given string is palindrome or not

#include<iostream>

#include<string>

#include<algorithm>

using namespace std;

inline bool isPalindrome(const string& str) {

string reversedStr = str;

reverse(reversedStr.begin(), reversedStr.end());

return str == reversedStr;

}

int main() {

string inputString;

cout << "Enter the string: ";

getline(cin, inputString);

if (isPalindrome(inputString)) {

cout << "THE GIVEN STRING IS PALINDROME" << endl;

} else {

cout << "THE GIVEN STRING IS NOT PALINDROME" << endl;

}

return 0;

}

Output:

Enter the string: MADAM

THE GIVEN STRING IS PALINDROME

19.Calculating simple interest

#include<iostream>

using namespace std;

double calculateSimpleInterest(double principal, int years, char isSeniorCitizen) {

double rateOfInterest = (isSeniorCitizen == 'y') ? 0.12 : 0.10;

// Calculating simple interest

double interest = principal \* rateOfInterest \* years;

return interest;

}

int main() {

// Input variables

double principal;

int years;

char isSeniorCitizen;

// User input

cout << "Enter the principal amount: ";

cin >> principal;

cout << "Enter the number of years: ";

cin >> years;

cout << "Is the customer a senior citizen (y/n): ";

cin >> isSeniorCitizen;

// Calculating and displaying the simple interest

double interest = calculateSimpleInterest(principal, years, isSeniorCitizen);

cout << "Interest: " << interest << endl;

return 0;

}

Output:

Enter the principal amount: 5000

Enter the number of years: 2

Is the customer a senior citizen (y/n): y

Interest: 1200

20.Eligible to vote or not

#include<iostream>

using namespace std;

// Function to get person's age from the user

int getPerson() {

int age;

cout << "Enter your age: ";

cin >> age;

return age;

}

// Function to check eligibility and display the result

void checkEligibility(int age) {

const int VOTING\_AGE = 18;

if (age >= VOTING\_AGE) {

cout << "You are eligible to vote." << endl;

} else {

int yearsLeft = VOTING\_AGE - age;

cout << "You are allowed to vote after " << yearsLeft << " years." << endl;

}

}

int main() {

// Get person's age from the user

int personAge = getPerson();

// Check eligibility and display the result

checkEligibility(personAge);

return 0;

}

Output:

Enter your age: 22

You are eligible to vote.

21.Username is valid or not

#include<iostream>

#include<string>

using namespace std;

// Function to check if the entered username is valid

bool isValidUsername(const string& username) {

// Add your validation criteria here

// For example, checking if the username contains a specific pattern

// In this example, the username should end with "@789"

size\_t found = username.find("@789");

return (found != string::npos && found == username.length() - 4);

}

int main() {

// Input variables

string username, reenteredUsername;

// User input

cout << "Enter the user name: ";

cin >> username;

cout << "Reenter the user name: ";

cin >> reenteredUsername;

// Checking if the entered usernames are valid

if (isValidUsername(username) && isValidUsername(reenteredUsername)) {

cout << "User name is valid." << endl;

} else {

cout << "User name is invalid." << endl;

}

return 0;

}

Output:

Enter the user name: 88970@pav

Reenter the user name: 88970@pav

User name is invalid.

22.Default arguments

#include<iostream>

using namespace std;

int calculateSum(int a = 0, int b = 0, int c = 0, int d = 0) {

return a + b + c + d;

}

int main() {

int value1, value2, value3, value4;

cout << "Enter the values: ";

cin >> value1 >> value2 >> value3 >> value4;

cout << "Sum: " << calculateSum(value1, value2, value3, value4) << endl;

return 0;

}

Output:

Enter the values:

10

20

41

25

Sum: 96

23.swapping of numbers

#include<iostream>

using namespace std;

void swapByValue(int x, int y) {

int temp = x;

x = y;

y = temp;

}

void swapByReference(int &x, int &y) {

int temp = x;

x = y;

y = temp;

}

int main() {

int a, b;

cout << "Enter the value for a: ";

cin >> a;

cout << "Enter the value for b: ";

cin >> b;

cout << "The value before swapping for call by value" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

swapByValue(a, b);

cout << "The value after swapping for call by value" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

swapByReference(a, b);

cout << "The value after swapping for call by reference" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

return 0;

}

Output:

Enter the value for a: 10

Enter the value for b: 20

The value before swapping for call by value

a = 10

b = 20

The value after swapping for call by value

a = 10

b = 20

The value after swapping for call by reference

a = 20

b = 10

24.Fibonacci series

#include<iostream>

using namespace std;

class FibonacciSeries {

private:

int limit;

public:

// Member function to get the limit from the user

void input() {

cout << "Enter the limit: ";

cin >> limit;

}

// Member function to print the Fibonacci series

void show() {

int a = 0, b = 1, c;

cout << "Fibonacci Series up to " << limit << ": ";

// Special case for 0

if (limit >= 1) {

cout << a << " ";

}

// Special case for 1

if (limit >= 2) {

cout << b << " ";

}

// Generating and printing the Fibonacci series

for (int i = 3; i <= limit; ++i) {

c = a + b;

cout << c << " ";

a = b;

b = c;

}

cout << endl;

}

};

int main() {

// Create an object of the FibonacciSeries class

FibonacciSeries fibSeries;

// Get the limit from the user

fibSeries.input();

// Display the Fibonacci series

fibSeries.show();

return 0;

}

Output:

Enter the limit: 5

Fibonacci Series up to 5: 0 1 1 2 3

25.Student grade

#include<iostream>

#include<string>

using namespace std;

class Student {

private:

string name;

int regNo;

int mark1, mark2, mark3;

float average;

public:

// Member function to input student details

void input() {

cout << "ENTER THE STUDENT NAME: ";

getline(cin, name);

cout << "ENTER THE REGISTER NUMBER: ";

cin >> regNo;

cout << "MARK 1: ";

cin >> mark1;

cout << "MARK 2: ";

cin >> mark2;

cout << "MARK 3: ";

cin >> mark3;

// Calculate average

average = (mark1 + mark2 + mark3) / 3.0;

}

// Member function to display average and grade

void display() {

cout << "AVERAGE SCORE IS: " << average << endl;

// Determine grade based on average

if (average > 90) {

cout << "GRADE: S" << endl;

} else if (average > 80) {

cout << "GRADE: A" << endl;

} else if (average > 70) {

cout << "GRADE: C" << endl;

} else if (average > 60) {

cout << "GRADE: D" << endl;

} else if (average > 50) {

cout << "GRADE: E" << endl;

} else {

cout << "GRADE: F" << endl;

}

}

};

int main() {

int numStudents;

// User input for the number of student entries

cout << "ENTER THE NUMBER OF STUDENT ENTRIES => ";

cin >> numStudents;

cin.ignore(); // Consume the newline character left by cin

// Array of Student objects

Student students[10];

// Input and display for each student

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

cout << "ENTER DETAILS FOR STUDENT " << i + 1 << ":" << endl;

students[i].input();

students[i].display();

cout << endl;

}

return 0;

}

Output:

ENTER THE NUMBER OF STUDENT ENTRIES => 1

ENTER DETAILS FOR STUDENT 1:

ENTER THE STUDENT NAME: Pavan

ENTER THE REGISTER NUMBER: 1992

MARK 1: 25

MARK 2: 45

MARK 3: 78

AVERAGE SCORE IS: 49.3333

GRADE: F

26.Binary operator overloading

#include<iostream>

using namespace std;

class Complex {

private:

int real;

int imag;

public:

// Constructor

Complex(int r = 0, int i = 0) : real(r), imag(i) {}

// Overloading binary operator '+'

Complex operator+(const Complex& other) {

Complex temp;

temp.real = real + other.real;

temp.imag = imag + other.imag;

return temp;

}

// Overloading binary operator '-'

Complex operator-(const Complex& other) {

Complex temp;

temp.real = real - other.real;

temp.imag = imag - other.imag;

return temp;

}

// Function to display complex number

void display() {

cout << real << " + " << imag << "i" << endl;

}

};

int main() {

// Input variables

int a, b;

// User input for the first complex number

cout << "Enter the value of Complex Numbers a, b: ";

cin >> a >> b;

Complex complex1(a, b);

// User input for the second complex number

cout << "Enter the value of Complex Numbers a, b: ";

cin >> a >> b;

Complex complex2(a, b);

// Displaying input values

cout << "Input Values" << endl;

complex1.display();

complex2.display();

// Adding and subtracting complex numbers

Complex resultAddition = complex1 + complex2;

Complex resultSubtraction = complex1 - complex2;

// Displaying results

cout << "Result" << endl;

resultAddition.display();

resultSubtraction.display();

return 0;

}

Output:

Enter the value of Complex Numbers a, b: 2

10

Enter the value of Complex Numbers a, b: 5

4

Input Values

2 + 10i

5 + 4i

Result

7 + 14i

-3 + 6i

27.income tax

#include<iostream>

using namespace std;

class IncomeTaxCalculator {

private:

double taxableIncome;

double tax;

public:

// Member function to get income from the user

void getIncome() {

cout << "Enter your Income: ";

cin >> taxableIncome;

}

// Member function to calculate income tax

void calculateTax() {

if (taxableIncome <= 60000) {

tax = 0;

} else if (taxableIncome > 60000 && taxableIncome <= 150000) {

tax = taxableIncome \* 0.05;

} else if (taxableIncome > 150000 && taxableIncome <= 500000) {

tax = taxableIncome \* 0.1;

} else {

tax = taxableIncome \* 0.15;

}

}

// Member function to display the calculated income tax

void displayTax() {

cout << "Your income tax is: " << tax << endl;

}

};

int main() {

// Create an object of the IncomeTaxCalculator class

IncomeTaxCalculator taxCalculator;

// Get income from the user

taxCalculator.getIncome();

// Calculate and display income tax

taxCalculator.calculateTax();

taxCalculator.displayTax();

return 0;

}

Output:

Enter your Income: 20000

Your income tax is: 0

28.