**EASY**

1. **Program**

#include <iostream>

using namespace std;

void display(char = '\*', int = 3);

int main() {

int count = 5;

cout << "No argument passed: ";

display();

cout << "First argument passed: ";

display('#');

cout << "Both arguments passed: ";

display('$', count);

return 0;

}

void display(char c, int count) {

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

{

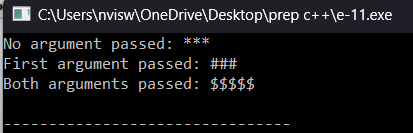
cout << c;

}

cout << endl;

}

**Output:**

****

1. **Program**

#include <iostream>

#include <string>

using namespace std;

bool isValidUsername(string username) {

if (username.empty()) {

return false;

}

if (username.length() > 15) {

return false;

}

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

if (!isalnum(username[i]) && username[i] != '\_') {

return false;

}

}

return true;

}

int main() {

string username;

cout << "Enter your username: ";

cin >> username;

if (isValidUsername(username)) {

cout << "The username is valid." << endl;

} else {

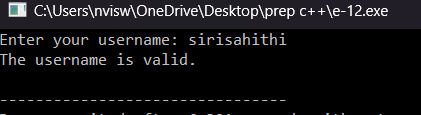
cout << "The username is invalid." << endl;

}

return 0;

}

**Output:**

****

1. **Program**

#include <iostream>

using namespace std;

int main() {

int age;

cout << "Enter your age: ";

cin >> age;

if (age >= 18) {

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

} else {

int yearsLeft = 18 - age;

cout << "You are not eligible to vote." << endl;

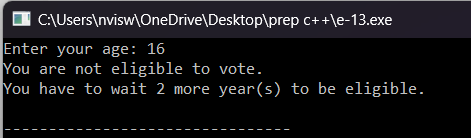
cout << "You have to wait " << yearsLeft << " more year(s) to be eligible." << endl;

}

return 0;

}

**Output:**

****

1. **Program**

#include <iostream>

using namespace std;

double SI(double p, int t, bool isSC) {

double r;

if (isSC) {

r = 12.0;

} else {

r = 10.0;

}

return (p \* r \* t) / 100.0;

}

int main() {

double p;

int t;

bool isSC;

char Response;

cout << "Enter the principal amount: ";

cin >> p;

cout << "Enter the time period (in years): ";

cin >> t;

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

cin >> Response;

isSC = (Response == 'y' || Response == 'Y');

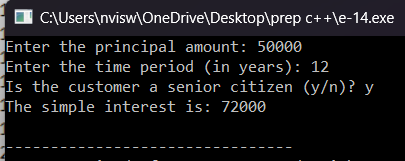
double interest = SI(p, t, isSC);

cout << "The simple interest is: " << interest << endl;

return 0;

}

**Output:**

****

1. **Program**

#include <iostream>

#include <string>

using namespace std;

inline bool isPalindrome(string str) {

int n = str.length();

for (int i = 0; i < n / 2; i++) {

if (str[i] != str[n - i - 1]) {

return false;

}

}

return true;

}

int main() {

string str;

cout << "Enter a string: ";

cin >> str;

if (isPalindrome(str)) {

cout << str << " is a palindrome." << endl;

} else {

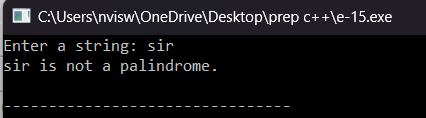
cout << str << " is not a palindrome." << endl;

}

return 0;

}

**Output:**

****

**Medium**

1. **Program**

#include <iostream>

using namespace std;

void display(char = '\*', int = 3);

int main() {

int count = 5;

cout << "No argument passed: ";

display();

cout << "First argument passed: ";

display('#');

cout << "Both arguments passed: ";

display('$', count);

return 0;

}

void display(char c, int count) {

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

{

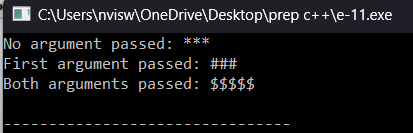
cout << c;

}

cout << endl;

}

**Output:**

****

**2.Program**

#include <iostream>

using namespace std;

int add(int a, int b) {

return a + b;

}

int add(int a, int b, int c) {

return a + b + c;

}

int main() {

int sum1 = add(10, 20);

int sum2 = add(12, 20, 23);

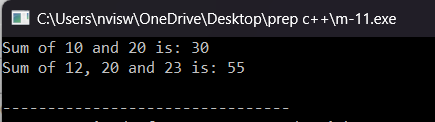
cout << "Sum of 10 and 20 is: " << sum1 << endl;

cout << "Sum of 12, 20 and 23 is: " << sum2 << endl;

return 0;

}

**Output:**

****

**3.Program**

#include <iostream>

using namespace std;

class Box {

private:

double width;

public:

double length;

void setWidth(double width) {

this->width = width;

}

double getWidth() {

return width;

}

};

int main() {

Box box;

box.length = 10;

box.setWidth(5);

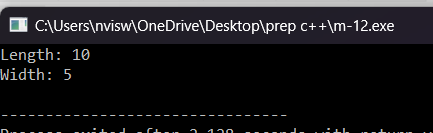
cout << "Length: " << box.length <<endl;

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

return 0;

}

**Output:**

****

**4. Program**

#include <iostream>

using namespace std;

#define R1 2

#define C1 2

#define R2 2

#define C2 2

void mulMat(int mat1[][C1], int mat2[][C2])

{

int rslt[R1][C2];

cout << "Multiplication of given two matrices is:\n";

for (int i = 0; i < R1; i++) {

for (int j = 0; j < C2; j++) {

rslt[i][j] = 0;

for (int k = 0; k < R2; k++) {

rslt[i][j] += mat1[i][k] \* mat2[k][j];

}

cout << rslt[i][j] << "\t";

}

cout << endl;

}

}

int main()

{

int mat1[R1][C1] = { { 1, 1 },

{ 2, 2 } };

int mat2[R2][C2] = { { 1, 1 },

{ 2, 2 } };

if (C1 != R2) {

cout << "The number of columns in Matrix-1 must "

"be equal to the number of rows in "

"Matrix-2"

<< endl;

cout << "Please update MACROs according to your "

"array dimension in #define section"

<< endl;

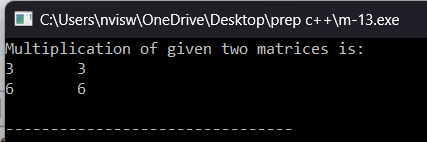
}

mulMat(mat1, mat2);

return 0;

}

**Output:**

****

**5. Program**

#include <iostream>

using namespace std;

class ObjectCounter {

private:

static int count;

public:

ObjectCounter() {

count++;

}

static int getCount() {

return count;

}

};

int ObjectCounter::count = 0;

int main() {

ObjectCounter obj1;

ObjectCounter obj2;

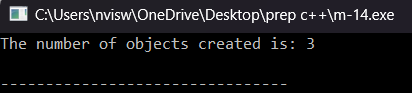
ObjectCounter obj3;

cout << "The number of objects created is: " << ObjectCounter::getCount() << endl;

return 0;

}

**Output:**

****

**HARD**

1. **Program**

#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 num1 = 5, num2 = 10;

cout << "Before swapping:" << endl;

cout << "num1 = " << num1 << ", num2 = " << num2 << endl;

swapByValue(num1, num2);

cout << "After swapping by value:" << endl;

cout << "num1 = " << num1 << ", num2 = " << num2 << endl;

num1 = 5;

num2 = 10;

swapByReference(num1, num2);

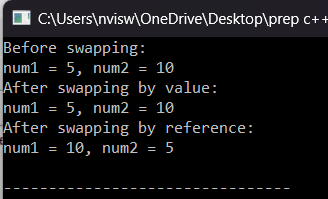
cout << "After swapping by reference:" << endl;

cout << "num1 = " << num1 << ", num2 = " << num2 << endl;

return 0;

}

**Output:**

****

1. **Program**

#include <iostream>

using namespace std;

class Series {

private:

int n1, n2, number;

public:

Series() {

n1 = 0;

n2 = 1;

}

void input() {

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

cin >> number;

}

void show() {

cout << "Fibonacci series:" << endl;

int next;

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

cout << n1 << " ";

next = n1 + n2;

n1 = n2;

n2 = next;

}

}

};

int main() {

Series obj;

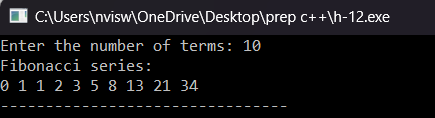
obj.input();

obj.show();

return 0;

}

**Output:**

****

**3. Program**

#include <iostream>

#include <string>

using namespace std;

class Student {

private:

string name;

int regNo;

double mark1, mark2, mark3;

double average;

char grade;

public:

void input() {

cout << "Enter student name: ";

getline(cin >> ws, name);

cout << "Enter registration number: ";

cin >> regNo;

cout << "Enter marks for three subjects: ";

cin >> mark1 >> mark2 >> mark3;

}

void calculateAverage() {

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

}

void calculateGrade() {

if (average > 90) {

grade = 'S';

} else if (average > 80) {

grade = 'A';

} else if (average > 70) {

grade = 'C';

} else if (average > 60) {

grade = 'D';

} else if (average > 50) {

grade = 'E';

} else {

grade = 'F';

}

}

void display() {

cout << "Name: " << name << endl;

cout << "Registration Number: " << regNo << endl;

cout << "Marks: " << mark1 << ", " << mark2 << ", " << mark3 << endl;

cout << "Average: " << average << endl;

cout << "Grade: " << grade << endl;

}

};

int main() {

Student student;

student.input();

student.calculateAverage();

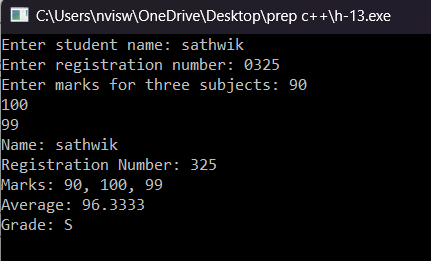
student.calculateGrade();

student.display();

return 0;

}

**Output:**

****

**4.Program**

#include <iostream>

using namespace std;

class Complex {

private:

float real;

float imag;

public:

Complex() { real = 0; imag = 0; }

Complex(float r, float i) { real = r; imag = i; }

Complex operator+(const Complex& other) {

Complex temp;

temp.real = real + other.real;

temp.imag = imag + other.imag;

return temp;

}

void print() {

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

}

};

int main() {

Complex c1(2, 3);

Complex c2(4, 5);

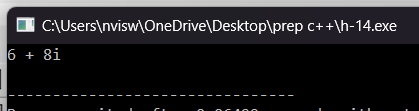
Complex c3 = c1 + c2;

c3.print();

return 0;

}

**Output:**

****

**5. Program**

#include <iostream>

using namespace std;

double calculateIncomeTax(double taxableIncome) {

double tax = 0;

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;

}

return tax;

}

int main() {

double taxableIncome;

cout << "Enter taxable income: ";

cin >> taxableIncome;

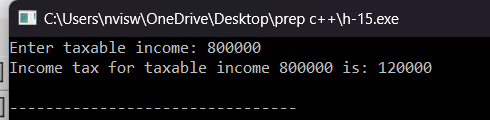
double tax = calculateIncomeTax(taxableIncome);

cout << "Income tax for taxable income " << taxableIncome << " is: " << tax << endl;

return 0;

}

**Output:**

****