Distance Converter:

Create a class named DistanceConverter. Include the following static methods:

convertMilesToKm(double miles): Converts miles to kilometers (1 mile = 1.60934 kilometers).

convertKmToMiles(double kilometers): Converts kilometers to miles. In your main function, prompt the user for a distance and a unit (miles or kilometers). Use the appropriate static method from the DistanceConverter class to perform the conversion and display the result to the user.

#include <iostream>

#include <string>

using namespace std;

class DistanceConverter {

public:

static double convertMilesToKm(double miles) {

return miles \* 1.60934;

}

static double convertKmToMiles(double kilometers) {

return kilometers / 1.60934;

}

};

int main() {

double distance;

string unit;

distance = 10.0;

unit = "miles";

if (unit == "miles") {

double kilometers = DistanceConverter::convertMilesToKm(distance);

printf("%.2f miles is equal to %.2f kilometers.\n", distance, kilometers);

} else if (unit == "kilometers") {

double miles = DistanceConverter::convertKmToMiles(distance);

printf("%.2f kilometers is equal to %.2f miles.\n", distance, miles);

} else {

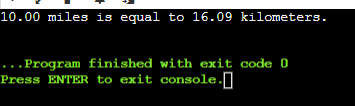
printf("Invalid unit. Please enter miles or kilometers.\n");

}

return 0;

}

Output:



Design a class named MathUtil. Include static methods for basic mathematical operations:

add(int a, int b): Adds two integers.

subtract(int a, int b): Subtracts two integers.

multiply(int a, int b): Multiplies two integers.

divide(int a, int b) (optional): Divides two integers with error handling for division by zero. In your main function, prompt the user for two numbers and an operation (+, -, \*, or /). Use the corresponding static method from the MathUtil class to perform the calculation and display the result.

#include <iostream>

#include <string>

using namespace std;

class MathUtil {

public:

static int add(int a, int b) {

return a + b;

}

static int subtract(int a, int b) {

return a - b;

}

static int multiply(int a, int b) {

return a \* b;

}

static double divide(int a, int b) {

if (b == 0) {

throw ("Division by zero is not allowed");

}

return static\_cast<double>(a) / b;

}

};

int main() {

int num1, num2;

char operation;

cout << "Enter the first number: ";

std::cin >> num1;

cout << "Enter the second number: ";

std::cin >> num2;

cout << "Enter the operation (+, -, \*, /): ";

std::cin >> operation;

try {

if (operation == '+') {

std::cout << "Result: " << MathUtil::add(num1, num2) << std::endl;

} else if (operation == '-') {

std::cout << "Result: " << MathUtil::subtract(num1, num2) << std::endl;

} else if (operation == '\*') {

std::cout << "Result: " << MathUtil::multiply(num1, num2) << std::endl;

} else if (operation == '/') {

std::cout << "Result: " << MathUtil::divide(num1, num2) << std::endl;

} else {

std::cout << "Invalid operation. Please enter +, -, \*, or /." << std::endl;

}

} catch (const std::runtime\_error& e) {

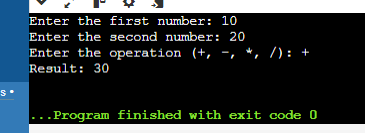
std::cout << "Error: " << e.what() << std::endl;

}

return 0;

}

Output:



Create a class named CurrencyConverter. Define a static variable named exchangeRate (e.g., USD to EUR exchange rate). Implement static methods:

convertToEur(double amount): Converts an amount from the base currency (USD) to EUR based on the exchange rate.

convertFromEur(double amount): Converts an amount from EUR to the base currency (USD). In your main function, prompt the user for an amount and a conversion direction (USD to EUR or EUR to USD). Use the appropriate static method from the CurrencyConverter class to perform the conversion and display the result.

#include <iostream>

using namespace std;

class CurrencyConverter {

public:

static double exchangeRate; // USD to EUR exchange rate

// Converts an amount from USD to EUR

static double convertToEur(double amount) {

return amount \* exchangeRate;

}

// Converts an amount from EUR to USD

static double convertFromEur(double amount) {

return amount / exchangeRate;

}

};

// Initialize the static variable

double CurrencyConverter::exchangeRate = 0.85; // Example exchange rate

int main() {

double amount;

char direction;

cout << "Enter the amount: ";

cin >> amount;

cout << "Enter the conversion direction (U for USD to EUR, E for EUR to USD): ";

cin >> direction;

if (direction == 'U' || direction == 'u') {

double convertedAmount = CurrencyConverter::convertToEur(amount);

cout << amount << " USD is " << convertedAmount << " EUR." << endl;

} else if (direction == 'E' || direction == 'e') {

double convertedAmount = CurrencyConverter::convertFromEur(amount);

cout << amount << " EUR is " << convertedAmount << " USD." << endl;

} else {

cout << "Invalid conversion direction entered." << endl;

}

return 0;

}

Output:

