1st question

To print callbyvalue and callbyrefrence

#include <iostream>

using namespace std;

void callByValue(int a, int b, int c) {

a += 10;

b += 20;

c += 30;

}

void callByReference(int &a, int &b, int &c) {

a += 10;

b += 20;

c += 30;

}

int main() {

int x, y, z;

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

cin >> x;

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

cin >> y;

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

cin >> z;

callByValue(x, y, z);

cout << " callByValue: x = " << x << ", y = " << y << ", z = " << z << endl;

callByReference(x, y, z);

cout << " callByReference: x = " << x << ", y = " << y << ", z = " << z << endl;

return 0;

}

2. polymorphism

#include <iostream>

using namespace std;

// Base class

class Animal {

public:

virtual void speak() { // Virtual function for polymorphism

cout << "Animal speaks" << endl;

}

};

// Derived class: Dog

class Dog : public Animal {

public:

void speak() override { // Override the base class method

cout << "Dog barks" << endl;

}

};

// Derived class: Cat

class Cat : public Animal {

public:

void speak() override { // Override the base class method

cout << "Cat meows" << endl;

}

};

int main() {

Animal\* animalPtr; // Base class pointer

Dog dog;

Cat cat;

// Pointing to Dog object

animalPtr = &dog;

animalPtr->speak(); // Calls Dog's speak()

// Pointing to Cat object

animalPtr = &cat;

animalPtr->speak(); // Calls Cat's speak()

return 0;

}

3. Sum of natural numbers

#include <iostream>

using namespace std;

int main() {

int a, b, sum = 0;

cout << "Enter the starting number: ";

cin >> a;

cout << "Enter the ending number: ";

cin >> b;

if (a > b) {

cout << "Starting number should be less than or equal to the ending number." << endl;

} else {

for (int i = a; i <= b; ++i) {

sum += i; // Add each number from a to b

}

cout << "The sum of numbers from " << a << " to " << b << " is: " << sum << endl;

}

return 0;

}

4. multiplication table

#include <iostream>

using namespace std;

int main() {

int number;

cout << "Enter any number to generate its multiplication table: ";

cin >> number;

cout << "Multiplication table of " << number << ":" << endl;

for (int i = 1; i <= 10; ++i) {

cout << number << " x " << i << " = " << number \* i << endl;

}

return 0;

}

5. reverse the string

#include<iostream>

#include<string>

using namespace std;

int main() {

string input;

cout << "Enter the sentence containing letters, digits and operators" << endl;

getline(cin, input);

cout << "The reversed sequence is:"<< endl;

for (int i = input.length() - 1; i >= 0; --i) {

cout << input[i];

}

cout << endl;

return 0;

}