

DOCUMENTATION RELATIVE VELOCITY

(description how to develop a program in c language)

Definition of Relative Velocity in One Dimension:

It measures how fast one object is moving relative to another in a straight line.

Formula :

If you have two objects, A and B, moving along the same straight line, with velocities v_A and v_B , respectively, the relative velocity of A with respect to B is given by:

$$V(A/B) = v_A - v_B;$$

Both Moving in the Same Direction:

- **Object A** is moving at 50 km/h.
- **Object B** is moving at 30 km/h.

The relative velocity of A with respect to B:

$$v_{A/B} = v_A - v_B$$

$$= 50 \text{ km/h} - 30 \text{ km/h}$$

$$= 20 \text{ km/h.}$$

Developing a C++ program using class:

Basic Terminologies :

Class :

A class is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

Advantages of class over structure:

- Polymorphism

Classes support both compile time and run time polymorphism, while structures do not.

- Inheritance

Classes support multiple and multilevel inheritance, while structures do not.

- Reference types

Classes are reference types, which makes passing a reference more efficient than passing a structure variable.

- Identity

Class instances have an identity and are passed by reference, while structs are handled as values.

- **Default accessibility**

Class members are private by default, while structure members are public by default.

Constructor:

In C++, a **constructor** is a special member function of a class that is automatically called when an object of that class is created. Its primary purpose is to initialize objects of the class. Constructors have the same name as the class and do not have a return type.

Benefits include automatic initialization, consistency, flexibility, simplified object creation, and encapsulation of initialization logic.

Types of constructors are:

- **Default Constructor:** Initializes objects without parameters.
- **Parameterized Constructor:** Initializes objects with specific values.
- **Copy Constructor:** Initializes objects as copies of other objects.
- **Move Constructor:** Moves resources from a temporary object to a new object.

PROGRAM :

INPUT:

```
> G+ relative_velocity.cpp > ...
#include <iostream>

class VelocityCalculator {
private:
    float velocity1;
    float velocity2;

public:
    // Constructor
    VelocityCalculator(float v1, float v2) : velocity1(v1), velocity2(v2) {}
    double relativeVelocity() const {
        return velocity1 - velocity2;
    }

    void displayVelocities() const {
        std::cout << "Velocity of object 1: " << velocity1 << " m/s" << std::endl;
        std::cout << "Velocity of object 2: " << velocity2 << " m/s" << std::endl;
    }
};

int main() {
    float v1, v2;
    std::cout << "Enter the velocity of object 1 (in m/s): ";
    std::cin >> v1;
    std::cout << "Enter the velocity of object 2 (in m/s): ";
    std::cin >> v2;

    VelocityCalculator calc(v1, v2);
    calc.displayVelocities();

    std::cout << "Relative velocity of object 1 with respect to object 2: "
        << calc.relativeVelocity() << " m/s" << std::endl;

    return 0;
}
```

Output:

```
PS C:\Users\Mona Bhatia\Desktop\GRC> cd "c:\Users\Mona Bhatia\Desktop\GRC\Program\" ; i
Enter the velocity of object 1 (in m/s): 44
Enter the velocity of object 2 (in m/s): 32
Velocity of object 1: 44 m/s
Velocity of object 2: 32 m/s
Relative velocity of object 1 with respect to object 2: 12 m/s
PS C:\Users\Mona Bhatia\Desktop\GRC\Program> █
```

“ Errors While developing program” :

So many common errors was there in this program:

→Syntax error Some speeling mistakes in the program

Semicolon forgot to put at the end;

Linker error is also happened, std:: direct it was telling the error.

