

**Assignment**

**Course Title : Object Oriented Programming**

**Course Code : CSE 121**

**Assignment No : 01**

Submitted By

Submitted To

Name : Musa Molla

ID : 21221203029

Program : B.Sc. Engg in CSE

Intake : 40 (DH)

Section : 01

**M. A. Nur Quraishi**

*Lecturer*

Department of Computer

Science & Engineering

Faculty Code: MANQ

Date of Submission

*07 October, 2024* 

Signature of Lecturer

#### 

#### 

## Objective:

The objective of this assignment is to implement a simple class hierarchy in C++, consisting of a base class Vehicle and derived classes Car, Truck, and Bike. We are also tasked with implementing a friend function middleSpeedChecker(), which checks whether the vehicle’s speed is within a given range. The function is to be tested in different contexts where it is made native to different classes.

### Scenario 1: Initial Class Design and Friend Function Implementation

#### Step 1: Class Design

I start by defining the base class Vehicle which has one protected attribute: speed. I then create three derived classes: Car, Truck, and Bike. Each of these derived classes has its own unique attribute:

* Car has deck
* Truck has container
* Bike has paddleGear

#### Step 2: Friend Function

The middleSpeedChecker() function will be a friend of the Car, Truck, and Bike classes. It will check if the speed of each vehicle is between 40 and 60 units, which is considered "medium" speed.

### Code for Scenario 1:

**#include<iostream>**

**using *namespace* std;**

***class* Vehicle {**

***protected:***

***int* speed;**

***public:***

**Vehicle(*int* *spd*) : speed(*spd*) {}**

**};**

***class* Car;**

***class* Truck;**

***class* Bike;**

***void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*, Bike*\** *b*);**

***class* Car : *public* Vehicle {**

***public:***

***int* deck;**

**Car(*int* *spd*, *int* *dk*) : Vehicle(*spd*), deck(*dk*) {}**

***friend* *void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*, Bike*\** *b*);**

**};**

***class* Truck : *public* Vehicle {**

***public:***

***int* container;**

**Truck(*int* *spd*, *int* *ctr*) : Vehicle(*spd*), container(*ctr*) {}**

***friend* *void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*, Bike*\** *b*);**

**};**

***class* Bike : *public* Vehicle {**

***public:***

***int* paddleGear;**

**Bike(*int* *spd*, *int* *gear*) : Vehicle(*spd*), paddleGear(*gear*) {}**

***friend* *void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*, Bike*\** *b*);**

**};**

***void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*, Bike*\** *b*) {**

**cout << "Checking middle speed for Car, Truck, and Bike..." << endl;**

**if (*c*->speed > 40 && *c*->speed < 60) {**

**cout << "Car is running at a medium speed." << endl;**

**} else {**

**cout << "Car is not running at a medium speed." << endl;**

**}**

**if (*t*->speed > 40 && *t*->speed < 60) {**

**cout << "Truck is running at a medium speed." << endl;**

**} else {**

**cout << "Truck is not running at a medium speed." << endl;**

**}**

**if (*b*->speed > 40 && *b*->speed < 60) {**

**cout << "Bike is running at a medium speed." << endl;**

**} else {**

**cout << "Bike is not running at a medium speed." << endl;**

**}**

**}**

***int* main() {**

**Car c(50, 4);**

**Truck t(55, 10);**

**Bike b(30, 2);**

**middleSpeedChecker(&c, &t, &b);**

**return 0;**

**}**

#### 

#### 

#### Output:

**m:\BUBT\teach-yourself-cpp-exercises\Midterm Assignment for CSE 111>**

**Checking middle speed for Car, Truck, and Bike...**

**Car is running at a medium speed.**

**Truck is running at a medium speed.**

**Bike is not running at a medium speed.**

### Scenario 2: middleSpeedChecker() Native to Car Class

In this scenario, we modify the middleSpeedChecker() function so that it becomes a member of the Car class, while remaining a friend of the Truck and Bike classes.

### Code for Scenario 2:

**#include<iostream>**

**using *namespace* std;**

***class* Vehicle {**

***protected:***

***int* speed;**

***public:***

**Vehicle(*int* *spd*) : speed(*spd*) {}**

**};**

***class* Truck;**

***class* Bike;**

***class* Car : *public* Vehicle {**

***public:***

***int* deck;**

**Car(*int* *spd*, *int* *dk*) : Vehicle(*spd*), deck(*dk*) {}**

***void* middleSpeedChecker(Truck*\** *t*, Bike*\** *b*);**

***friend* *class* Truck;**

***friend* *class* Bike;**

**};**

***class* Truck : *public* Vehicle {**

***public:***

***int* container;**

**Truck(*int* *spd*, *int* *ctr*) : Vehicle(*spd*), container(*ctr*) {}**

***friend* *class* Car;**

***friend* *class* Bike;**

**};**

***class* Bike : *public* Vehicle {**

***public:***

***int* paddleGear;**

**Bike(*int* *spd*, *int* *gear*) : Vehicle(*spd*), paddleGear(*gear*) {}**

***friend* *class* Car;**

***friend* *class* Truck;**

**};**

***void* Car::middleSpeedChecker(Truck*\** *t*, Bike*\** *b*) {**

**cout << "Checking middle speed for Car, Truck, and Bike..." << endl;**

**if (speed > 40 && speed < 60) {**

**cout << "Car is running at a medium speed." << endl;**

**} else {**

**cout << "Car is not running at a medium speed." << endl;**

**}**

**if (*t*->speed > 40 && *t*->speed < 60) {**

**cout << "Truck is running at a medium speed." << endl;**

**} else {**

**cout << "Truck is not running at a medium speed." << endl;**

**}**

**if (*b*->speed > 40 && *b*->speed < 60) {**

**cout << "Bike is running at a medium speed." << endl;**

**} else {**

**cout << "Bike is not running at a medium speed." << endl;**

**}**

**}**

***int* main() {**

**Car c(50, 4);**

**Truck t(55, 10);**

**Bike b(65, 2);**

**c.middleSpeedChecker(&t, &b);**

**return 0;**

**}**

#### Output:

**m:\BUBT\teach-yourself-cpp-exercises\Midterm Assignment for CSE 111>**

**Checking middle speed for Car, Truck, and Bike...**

**Car is running at a medium speed.**

**Truck is running at a medium speed.**

**Bike is not running at a medium speed.**

### Scenario 3: middleSpeedChecker() Native to Bike Class

Next, we modify the middleSpeedChecker() function so that it becomes a member of the Bike class and remains a friend to the Car and Truck classes.

### Code for Scenario 3:

**#include<iostream>**

**using *namespace* std;**

***class* Vehicle {**

***protected:***

***int* speed;**

***public:***

**Vehicle(*int* *spd*) : speed(*spd*) {}**

**};**

***class* Car;**

***class* Truck;**

***class* Bike : *public* Vehicle {**

***public:***

***int* paddleGear;**

**Bike(*int* *spd*, *int* *gear*) : Vehicle(*spd*), paddleGear(*gear*) {}**

***void* middleSpeedChecker(Car*\** *c*, Truck*\** *t*);**

***friend* *class* Car;**

***friend* *class* Truck;**

**};**

***class* Car : *public* Vehicle {**

***public:***

***int* deck;**

**Car(*int* *spd*, *int* *dk*) : Vehicle(*spd*), deck(*dk*) {}**

***friend* *class* Bike;**

***friend* *class* Truck;**

**};**

***class* Truck : *public* Vehicle {**

***public:***

***int* container;**

**Truck(*int* *spd*, *int* *ctr*) : Vehicle(*spd*), container(*ctr*) {}**

***friend* *class* Car;**

***friend* *class* Bike;**

**};**

***void* Bike::middleSpeedChecker(Car*\** *c*, Truck*\** *t*) {**

**cout << "Checking middle speed for Bike, Car, and Truck..." << endl;**

**if (speed > 40 && speed < 60) {**

**cout << "Bike is running at a medium speed." << endl;**

**} else {**

**cout << "Bike is not running at a medium speed." << endl;**

**}**

**if (*c*->speed > 40 && *c*->speed < 60) {**

**cout << "Car is running at a medium speed." << endl;**

**} else {**

**cout << "Car is not running at a medium speed." << endl;**

**}**

**if (*t*->speed > 40 && *t*->speed < 60) {**

**cout << "Truck is running at a medium speed." << endl;**

**} else {**

**cout << "Truck is not running at a medium speed." << endl;**

**}**

**}**

***int* main() {**

**Bike b(30, 2);**

**Car c(50, 4);**

**Truck t(55, 10);**

**b.middleSpeedChecker(&c, &t);**

**return 0;**

**}**

#### Output:

**m:\BUBT\teach-yourself-cpp-exercises\Midterm Assignment for CSE 111>**

**Checking middle speed for Bike, Car, and Truck...**

**Bike is not running at a medium speed.**

**Car is running at a medium speed.**

**Truck is running at a medium speed.**

### Scenario 4: middleSpeedChecker() Native to Truck Class

Finally, we modify the middleSpeedChecker() function to be native to the Truck class and friend to the other two classes (Car and Bike).

### Code for Scenario 4:

**#include<iostream>**

**using *namespace* std;**

***class* Vehicle {**

***protected:***

***int* speed;**

***public:***

**Vehicle(*int* *spd*) : speed(*spd*) {}**

**};**

***class* Car;**

***class* Bike;**

***class* Truck : *public* Vehicle {**

***public:***

***int* container;**

**Truck(*int* *spd*, *int* *ctr*) : Vehicle(*spd*), container(*ctr*) {}**

***void* middleSpeedChecker(Car*\** *c*, Bike*\** *b*);**

***friend* *class* Car;**

***friend* *class* Bike;**

**};**

***class* Car : *public* Vehicle {**

***public:***

***int* deck;**

**Car(*int* *spd*, *int* *dk*) : Vehicle(*spd*), deck(*dk*) {}**

***friend* *class* Truck;**

***friend* *class* Bike;**

**};**

***class* Bike : *public* Vehicle {**

***public:***

***int* paddleGear;**

**Bike(*int* *spd*, *int* *gear*) : Vehicle(*spd*), paddleGear(*gear*) {}**

***friend* *class* Truck;**

***friend* *class* Car;**

**};**

***void* Truck::middleSpeedChecker(Car*\** *c*, Bike*\** *b*) {**

**cout << "Checking middle speed for Truck, Car, and Bike..." << endl;**

**if (speed > 40 && speed < 60) {**

**cout << "Truck is running at a medium speed." << endl;**

**} else {**

**cout << "Truck is not running at a medium speed." << endl;**

**}**

**if (*c*->speed > 40 && *c*->speed < 60) {**

**cout << "Car is running at a medium speed." << endl;**

**} else {**

**cout << "Car is not running at a medium speed." << endl;**

**}**

**if (*b*->speed > 40 && *b*->speed < 60) {**

**cout << "Bike is running at a medium speed." << endl;**

**} else {**

**cout << "Bike is not running at a medium speed." << endl;**

**}**

**}**

***int* main() {**

**Truck t(10, 10);**

**Car c(50, 4);**

**Bike b(50, 2);**

**t.middleSpeedChecker(&c, &b);**

**return 0;**

**}**

#### 

#### Output:

**m:\BUBT\teach-yourself-cpp-exercises\Midterm Assignment for CSE 111>**

**Checking middle speed for Truck, Car, and Bike...**

**Truck is not running at a medium speed.**

**Car is running at a medium speed.**

**Bike is running at a medium speed.**

### Conclusion:

We successfully implemented the class hierarchy, created a middleSpeedChecker() function, and tested it in four different contexts. We demonstrated how the function behaves when made a friend function and when it is native to different classes (Car, Bike, and Truck). Each scenario provided insights into class inheritance, friend functions, and encapsulation principles in C++.