# Task 3:

**Header File:**

#include<iostream>

#include<string>

using namespace std;

class Person

{

public:

string name;

int age;

string gender;

Person();

//Setter FUnctions

void namesetter(string Name);

void agesetter(int age);

void gendersetter(string gender);

//Getter Functions

string namegetter();

int agegetter();

string gendergetter();

};

class EmployedPerson : public Person

{

public:

int nic;

EmployedPerson();

//Setter FUnctions

void nicsetter(int number);

//Getter Functions

int nicgetter();

void employe();

};

class UnemployedPerson : public Person

{

public:

UnemployedPerson();

};

class BusinessMan : public EmployedPerson

{

public:

BusinessMan();

void display();

};

**FUNCTION FILE:**

#pragma once

#include "Header.h"

//Setter Definition

Person::Person(){}

void Person :: namesetter(string Name)

{

this->name = Name;

}

void Person :: agesetter(int age)

{

this->age = age;

}

void Person :: gendersetter(string gender)

{

this->gender = gender;

}

//Getter Definition

string Person :: namegetter()

{

return name;

}

int Person :: agegetter()

{

return age;

}

string Person :: gendergetter()

{

return gender;

}

//Setter Definition

EmployedPerson :: EmployedPerson(){}

void EmployedPerson :: nicsetter(int number)

{

this->nic = number;

}

//Getter Definition

int EmployedPerson :: nicgetter()

{

return nic;

}

void EmployedPerson :: employe()

{

cout << "HI! I am employe from Employed Class." << endl;

}

UnemployedPerson :: UnemployedPerson()

{

cout << "Hi, I am UnEmploy from UnEmployed Class" << endl;

}

BusinessMan :: BusinessMan(){}

void BusinessMan :: display()

{

cout << "Name: " << namegetter() << endl;

cout << "Age: " << agegetter() << endl;

cout << "Gender: " << gendergetter() << endl;

cout << "NIC: " << nicgetter() << endl;

}

**SOURCE FILE:**

#include"Header.h"

int main()

{

BusinessMan b1;

b1.employe();

b1.namesetter("Huzaifa");

b1.agesetter(20);

b1.gendersetter("Male");

b1.nicsetter(123456789);

b1.display();

UnemployedPerson up1;

up1.namesetter("Ali");

up1.agesetter(22);

up1.gendersetter("Male");

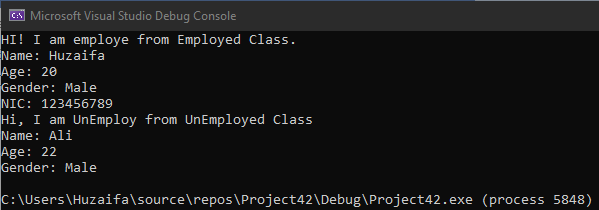
cout << "Name: " << up1.name << endl;

cout << "Age: " << up1.age << endl;

cout << "Gender: " << up1.gender << endl;

}

# Output:



# Task 4:

**Header File:**

#include<iostream>

#include<string>

using namespace std;

class Vehicle

{

public:

int speed;

int distance;

Vehicle();

//Setter FUnctions

void speedsetter(int speed);

void distancesetter(int dist);

//Getter Functions

int speedgetter();

int distancegetter();

};

class WheelVehicle : public Vehicle

{

public:

int wheels;

WheelVehicle();

//Setter FUnctions

void wheelsetter(int wheel);

//Getter Functions

int wheelgetter();

};

class WingVehicle : public Vehicle

{

public:

int wings;

WingVehicle();

//Setter FUnctions

void wingsetter(int wing);

//Getter Functions

int winggetter();

};

class Truck : public WheelVehicle

{

public:

Truck();

int carryingLoad;

//Setter FUnctions

void loadsetter(int load);

//Getter Functions

int loadgetter();

};

**FUNCTION FILE:**

#pragma once

#include "Header.h"

Vehicle::Vehicle() {}

//Setter Definition

void Vehicle:: speedsetter(int speed)

{

this->speed = speed;

}

void Vehicle:: distancesetter(int dist)

{

this->distance = dist;

}

//Getter Definition

int Vehicle:: speedgetter()

{

return speed;

}

int Vehicle:: distancegetter()

{

return distance;

}

WheelVehicle ::WheelVehicle() {}

//Setter Definition

void WheelVehicle:: wheelsetter(int wheel)

{

this->wheels = wheel;

}

//Getter Definition

int WheelVehicle:: wheelgetter()

{

return wheels;

}

WingVehicle :: WingVehicle(){}

//Setter Definition

void WingVehicle :: wingsetter(int wing)

{

this->wings = wing;

}

//Getter Definition

int WingVehicle :: winggetter()

{

return wings;

}

Truck :: Truck() {}

//Setter Definition

void Truck :: loadsetter(int load)

{

this->carryingLoad = load;

}

//Getter Definition

int Truck :: loadgetter()

{

return carryingLoad;

}

**SOURCE FILE:**

#include"Header.h"

int main()

{

Vehicle v1;

v1.speedsetter(40);

v1.distancesetter(500);

WheelVehicle v2;

v2.wheelsetter(4);

v2.speedsetter(120);

v2.distancesetter(600);

WingVehicle v3;

v3.wingsetter(2);

v3.speedsetter(4000);

v3.distancesetter(5000);

Truck v4;

v4.loadsetter(1000);

v4.wheelsetter(10);

v4.speedsetter(100);

v4.distancesetter(800);

cout << "VEHICLE 1: " << endl;

cout << "Speed: " << v1.speedgetter() << endl;

cout << "Distance: " << v1.distancegetter() << endl << endl;

cout << "VEHICLE 2: " << endl;

cout << "Wheels: " << v2.wheelgetter() << endl;

cout << "Speed: " << v2.speedgetter() << endl;

cout << "Distance: " << v2.distancegetter() << endl << endl;

cout << "VEHICLE 3: " << endl;

cout << "Wing: " << v3.winggetter() << endl;

cout << "Speed: " << v3.speedgetter() << endl;

cout << "Distance: " << v3.distancegetter() << endl << endl;

cout << "VEHICLE 4: " << endl;

cout << "Load: " << v4.loadgetter() << endl;

cout << "Wheels: " << v4.wheelgetter() << endl;

cout << "Speed: " << v4.speedgetter() << endl;

cout << "Distance: " << v4.distancegetter() << endl << endl;

}

# Output:

