//#include <iostream>

//using namespace std;

//

//class Person

//{

// string name;

// int age;

//public:

// Person() : name(""), age(0) {}

// Person(string name, int age)

// {

// this->name = name;

// this->age = age;

// }

//

// int GetBirthYear() const

// {

// return 2022 - age;

// }

// void Show() const

// {

// cout << " === PERSON INFO === " << endl;

// cout << " Name : " << name << endl;

// cout << " Age : " << age << endl;

// }

//};

//

//class Employee :public Person

//{

// int id;

// int salary;

//public:

// Employee() : id(0),salary(0){}

// Employee(string name,int age, int id, int salary)

// :Person(name,age)

// {

// this->id = id;

// this->salary = salary;

// }

// //fake override

// void Show() const

// {

// Person::Show();

// cout << " Birth year : " << GetBirthYear() << endl;

// cout << " ID : " << id << endl;

// cout << " Salary : " << salary << endl;

// }

//};

//

//class Supervisor :public Employee

//{

// string department;

// int employees[5];

//public:

// Supervisor(string name,int age,int id, int salary,string department)

// :Employee(name,age,id,salary)

// {

// this->department = department;

// for (int i = 0; i < 5; i++)

// {

// employees[i] = i + 1000;

// }

// }

//

// void Show()const

// {

// Employee::Show();

// cout << " Department : " << department << endl;

// cout << " Employees id : " << endl;

// for (int x = 0; x < 5; x++)

// {

// cout << " " << employees[x];

// }

// }

//};

//

//int main()

//{

// //Employee e("John",42,1,1200);

// //e.Show();

//

// Supervisor tofiq("Tofiq", 56, 1, 2300, "IT Department");

// tofiq.Show();

// return 0;

//}

#include <iostream>

using namespace std;

class Bird

{

string nickname;

string color;

bool hasFeathers;

public:

Bird() : nickname(""), color(""),hasFeathers(false) {}

Bird(const string& name, const string& color, const bool& hasFeathers)

{

this->nickname = name;

this->color = color;

this->hasFeathers = hasFeathers;

}

void Show() const

{

cout << "\n ======= BIRD ======= " << endl;

cout << " Nickname : " << nickname << endl;

cout << " Color : " << color << endl;

cout << " Has feathers : ";

if (hasFeathers)

{

cout << " Yes " << endl;

}

else

{

cout << " No " << endl;

}

}

};

// can fly

class Fly : public Bird

{

int maxFlySpeed;

int maxHeight;

public:

Fly() : maxFlySpeed(0), maxHeight(0){}

Fly(const string& name, const string& color, const bool& hasFeathers,const int& maxFlySpeed, const int& maxHeight)

:Bird(name,color,hasFeathers)

{

this->maxFlySpeed = maxFlySpeed;

this->maxHeight = maxHeight;

}

void Show() const

{

Bird::Show();

cout << " Max Fly Speed : " << maxFlySpeed << endl;

cout << " Max Height : " << maxHeight << endl;

}

};

// cannot fly

class Nonfly : public Bird

{

int maxRunSpeed;

int weight;

public:

Nonfly() : maxRunSpeed(0), weight(0) {}

Nonfly(const string& name, const string& color, const bool& hasFeathers, const int& maxRunSpeed, const int& weight)

:Bird(name, color, hasFeathers)

{

this->maxRunSpeed = maxRunSpeed;

this->weight = weight;

}

void Show() const

{

Bird::Show();

cout << " Max Run Speed : " << maxRunSpeed << endl;

cout << " Weight : " << weight << endl;

}

};

class Swallow : public Fly

{

bool hasEyes;

public:

Swallow() : hasEyes(false) {}

Swallow(const string& name, const string& color, const bool& hasFeathers, const int& maxFlySpeed, const int& maxHeight, const bool& hasEyes)

:Fly(name, color, hasFeathers, maxFlySpeed, maxHeight)

{

this->hasEyes = hasEyes;

}

void Show() const

{

Fly::Show();

cout << " Has Eyes : ";

if (hasEyes)

{

cout << " Yes " << endl;

}

else

{

cout << " No " << endl;

}

}

};

class Eagle : public Fly

{

int age;

public:

Eagle() : age(0) {}

Eagle(const string& name, const string& color, const bool& hasFeathers, const int& maxFlySpeed, const int& maxHeight,const int& age)

:Fly(name,color,hasFeathers,maxFlySpeed,maxHeight)

{

this->age = age;

}

void Show() const

{

Fly::Show();

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

}

};

class Camelbird :public Nonfly

{

bool hasLegs;

public:

Camelbird() : hasLegs(false) {}

Camelbird(const string& name, const string& color, const bool& hasFeathers, const int& maxRunSpeed, const int& weight, const bool& hasLegs)

:Nonfly(name, color, hasFeathers, maxRunSpeed, weight)

{

this->hasLegs = hasLegs;

}

void Show() const

{

Nonfly::Show();

cout << " Has Legs : ";

if (hasLegs)

{

cout << " Yes " << endl;

}

else

{

cout << " No " << endl;

}

}

};

class Penguin : public Nonfly

{

bool hasBrain;

public:

Penguin() : hasBrain(false) {}

Penguin(const string& name, const string& color, const bool& hasFeathers, const int& maxRunSpeed, const int& weight, const bool& hasBrain)

:Nonfly(name, color, hasFeathers, maxRunSpeed, weight)

{

this->hasBrain = hasBrain;

}

void Show() const

{

Nonfly::Show();

cout << " Has Brain : ";

if (hasBrain)

{

cout << " Yes " << endl;

}

else

{

cout << " No " << endl;

}

}

};

int main()

{

Swallow swallow("swallow", "red", true, 120, 400, true);

swallow.Show();

Eagle eagle("eagle", "black", true, 300, 1000, 30);

eagle.Show();

Camelbird camelbird("camelbird", "brown", false, 30, 100, true);

camelbird.Show();

Penguin penguin("penguin", "white", false, 12, 40, true);

penguin.Show();

return 0;

}

#include <iostream>

using namespace std;

//// Multiple Inheritance

//class SMS

//{

//public:

// int id;

// string text;

// string number;

// SMS()

// {

// cout << " SMS default constructor " << endl;

// id = 1;

// text = "Hello guys";

// number = "12312421";

// }

//

// void Show() const

// {

// cout << " ========= SMS ========= " << endl;

// cout << id << endl;

// cout << text << endl;

// cout << number << endl;

// }

//};

//

//class Email

//{

//public:

// int id;

// string text;

// string ToEmail;

// Email()

// {

// cout << " EMAIL default constructor " << endl;

// id = 1;

// text = "Hello, I wrote from email";

// ToEmail = "John";

// }

//

// void Show() const

// {

// cout << " ========= EMAIL ========= " << endl;

// cout << id << endl;

// cout << text << endl;

// cout << "to " << ToEmail << endl;

// }

//};

//

//class MessageService:public SMS,public Email // 1ci SMSin sonra emailin constructoru yaranir

//{

// string companyName;

//public:

// MessageService()

// {

// companyName = "babat shirket";

// }

//

// void Show() const

// {

// //cout << "SMS text : " << text << endl; // text is ambigious

// cout << "SMS text : " << SMS::text << endl;

// cout << "Email text : " << Email::text << endl;

// }

//};

//

//

//int main()

//{

// MessageService ms;

// ms.Show();

//

// return 0;

//}

class A

{

public:

A()

{

cout << " Default constructor of A " << endl;

}

};

class B: virtual public A

{

};

class C : virtual public A

{

};

class D : public B, public C

{

};

int main()

{

D d;

return 0;

}