#include<iostream>

using namespace std;

//class Base {

//protected:

// int value;

//public:

// Base(int value) {

// this->value = value;

// }

// int getValue()const {

// return value;

// }

// virtual string GetName() {

// return "i am base";

// }

//};

//

//

//class Derived :public Base {

//public:

// Derived(int value):Base(value){}

//

// int GetDoubleValue() {

// return value \* 2;

// }

//

// string GetName() {

// return "i am derived";

// }

//

//};

//

//

//void main() {

//

// /\*Base base(10);

// cout << base.GetName() << endl;

// Derived derived(20);

// cout << derived.GetName() << endl;\*/

// //Derived derived(20);

// Base\* base = new Derived(20);

//

// cout << base->GetName() << endl;

//}

//class Animal {

//protected:

// string name;

// Animal(string name) {

// this->name = name;

// }

//

//public:

// string GetName()const {

// return name;

// }

// virtual void Speak() {

// cout << "?????????????" << endl;

// }

//};

//

//

//class Cat :public Animal {

//public:

// Cat(string name):Animal(name){}

// void Speak() override final {

// cout << "meaouououou" << endl;

// }

//

// void Run() {

// cout << "i can run like cat" << endl;

// }

//};

//

//

//class Tiger :public Cat {

//public:

// Tiger(string name):Cat(name){}

//

//

//

// };

//

//class Dog :public Animal {

//public:

// Dog(string name):Animal(name){}

// void Speak() {

// cout << "How how how , i can bark like real dog" << endl;

// }

// void BarkBark() {

// cout << "i can just bark" << endl;

// }

//};

//

//

//

//class Elephant :public Animal {

//public:

// Elephant(string name):Animal(name){}

//

//

// void Speak() {

// cout << "Bruwwwww , men filem" << endl;

// }

//};

//

//

//void main() {

// Animal\* dog = new Dog("");

// Animal\* cat = new Cat("");

// Animal\* tiger = new Tiger("");

// dog->Speak();

// cat->Speak();

// tiger->Speak();

//}

//void CallMethod(Animal\* animal) {

// animal->Speak();

//}

//void Show() {

// cout << 10 << endl;

//}

//

//void add(int a, int b) {

// cout << a + b << endl;

//}

//

//void main() {

//

// Animal\* cat = new Cat("Mestan");

// Animal\* dog = new Dog("Qarabash");

// Animal\* elephant = new Elephant("Fred");

//

//

// Animal\*\* animals = new Animal \* [3]{ cat,dog,elephant };

//

// for (size\_t i = 0; i < 3; i++)

// {

// animals[i]->Speak();

// }

//

// //CallMethod(elephant);

// //cat->Speak();

// //dog->Speak();

// //elephant->Speak();

//

// //Early Binding ve ya Static Polymorphism

// //Show();

// //add(10, 20);

//

// /\*void(\*ptr)(int, int) = add;

// ptr(10,20);\*/

// //Late Binding ve ya Dynamic Polymophism

//

//}

//

class Human {

protected:

string name;

int age;

Human(string name, int age) :name(name), age(age) {}

public:

virtual void Run() {

cout << "My Speed is not good" << endl;

}

virtual ~Human()

{

cout << "Human Destructor" << endl;

}

};

class SimpleHuman :public Human {

public:

SimpleHuman(string name, int age) :Human(name, age) {

}

~SimpleHuman()

{

cout << "Destructor Simple Human" << endl;

}

};

class UsainBolt :public Human {

public:

UsainBolt(string name, int age) :Human(name, age) {}

void Run()override final {

cout << "my speed is awesome" << endl;

}

~UsainBolt()

{

cout << "Destructor Usain Bolt" << endl;

}

};

void main() {

// Human\* h1 = new SimpleHuman("Rafiq", 34);

// h1->Run();

Human\* h2 = new UsainBolt("Huseyn", 30);

// h2->Run();

delete h2;

}