Design a base class Animal with a pure virtual function makeSound() that returns a string representing the animal's sound. Derive classes like Dog, Cat, and Bird from Animal, each overriding makeSound() with the appropriate sound ("Woof!", "Meow!", "Chirp!"). Create a function playAnimalSound(Animal\* animal) that takes an Animal pointer and calls makeSound(). Populate an Animal\* array with various animal objects and use playAnimalSound() to hear their sounds polymorphically.

#include <iostream>

#include <vector>

#include <string>

class Animal {

public:

virtual std::string makeSound() const = 0;

virtual ~Animal() {}

};

class Dog : public Animal {

public:

std::string makeSound() const override {

return "Woof!";

}

};

class Cat : public Animal {

public:

std::string makeSound() const override {

return "Meow!";

}

};

class Bird : public Animal {

public:

std::string makeSound() const override {

return "Chirp!";

}

};

void playAnimalSound(const Animal\* animal) {

std::cout << animal->makeSound() << std::endl;

}

int main() {

Dog dog;

Cat cat;

Bird bird;

std::vector<Animal\*> animals = {&dog, &cat, &bird};

for(const auto& animal : animals) {

playAnimalSound(animal);

}

return 0;

}

Output:

