Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

(Base and Derived classes should already be coded in part 3.

Create an animal base class.

```
string animalName;

string sound   // Use constructors to set appropriate derived class sound...

virtual animalSound()            { cout << AnimalName << " says " << sound << endl; }
```

Create 4 new derived classes based on animal, that override the virtual function, animalSound, with the specific animal noise,

and assign the animal name to the base animal name variable.

cat

dog

elephant

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

mouse

Declare instances of each of the 4 different animals. C1, D1, E1, M1.

Write a function that has one parameter, a derived pointer

The function void polyNoise(baseClass * ptrBase) { cout ptrBase->animalSound;}

//Note polymorphis - This one line of code should work for ALL/Any of the different derived classes.

Call the function 4 times, pass each animal derived pointer to it and the output should be the correct animal sound.

```
//
//  main.cpp
//  Week 14 Prog 4
//
//  Created by Pippo Pesic on 11/17/22.
```

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

//

```
#include <iostream>
#include <string>
using namespace std;

class Animal {
	string animalName;
	string sound;
public:
	Animal(string n, string s) {
	animalName = n;
	sound = s;
	}
	virtual void animalSound() {
	cout << animalName << " says " << sound << endl;
	};
};

class Cat : public Animal {
public:
```

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

```
        Cat () : Animal("cat", "meow") {}
};

class Dog : public Animal {
public:
        Dog () : Animal("dog", "bark") {}
};

class Elephant : public Animal {
public:
        Elephant () : Animal("elephant", "squeal") {}
};

class Mouse : public Animal {
public:
        Mouse () : Animal("mouse", "squeak") {}
};

void polyNoise(Animal *pAnimal) {
        pAnimal->animalSound();
};
```

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

```
int main() {
        char animal;
        Cat c1;
        Dog d1;
        Elephant e1;
        Mouse m1;

        Animal *pAnimal = NULL;

        cout << "Select an Animal (C, D, E, M) - ";
        cin >> animal;

        switch (animal) {
        case 'C':
        pAnimal = &c1;
        break;
        case 'D':
        pAnimal = &d1;
        break;
        case 'E':
```

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4

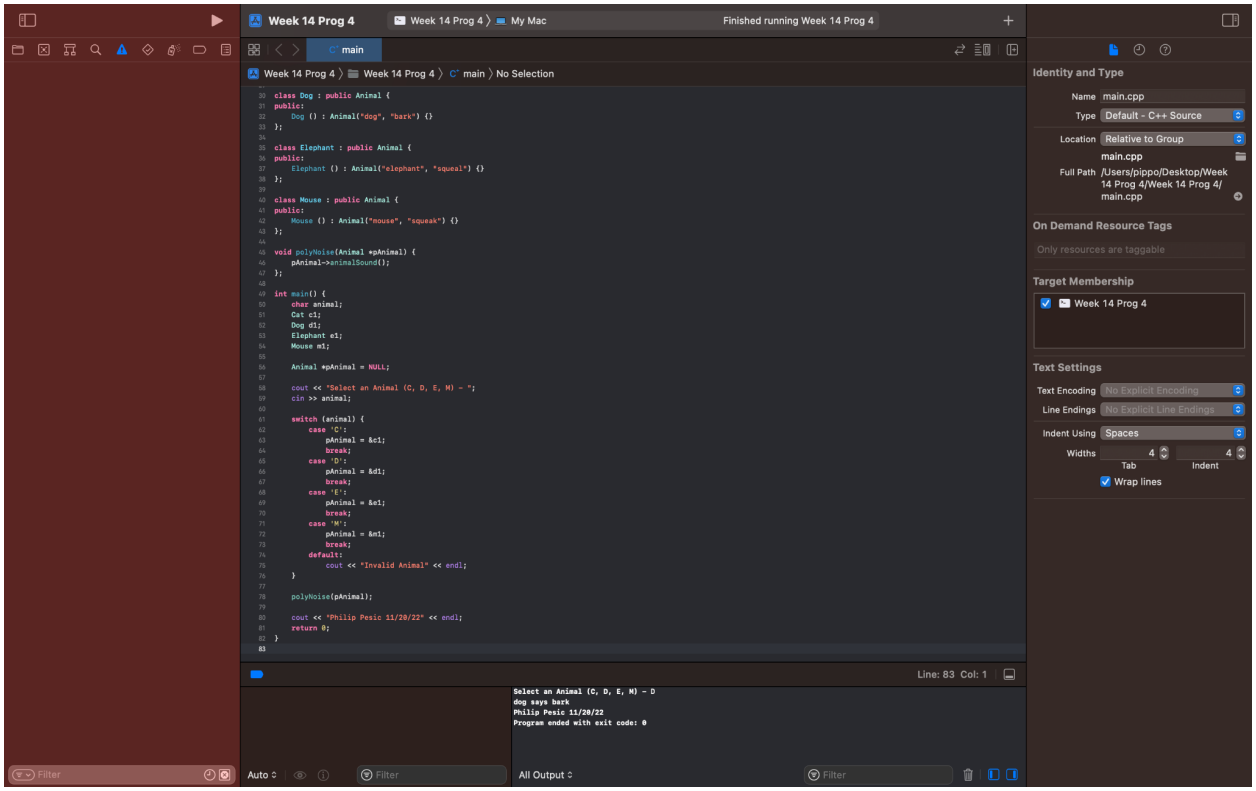
```
        pAnimal = &e1;
        break;
        case 'M':
        pAnimal = &m1;
        break;
        default:
        cout << "Invalid Animal" << endl;
        }

        polyNoise(pAnimal);

        cout << "Philip Pesic 11/20/22" << endl;
        return 0;
}
```

Philip Pesic

Week 14

November 20 2022

Week 14 Prog 4



I learned: how to create a polymorphic constructor that calls a base constructor