

Activity 12

Comp 11 - Summer Session — Inheritance

With a partner(or two), discuss the following code sample and answer the questions below. The instructor and teaching assistants will let you discuss and then be around to answer questions. ¹

12.1 Description

For this activity, take a look at the following code. It shows off some features of inheritance.

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4
5 // =====
6 // ==Base Class from which all animals inherit from==
7 // =====
8 class Animal{
9 public:
10     virtual ~Animal() { }
11
12     void getName(){
13         std::cout << "Hello, my name is: " << name << "\n";
14     }
15
16     void setName(std::string _name) {
17         name = _name;
18     }
19
20 private:
21     std::string name;
22 };
23
24 // =====
25 // == All Cat's use Animal as a Base Class ==
```

¹Activities do not need to be returned to instructors, they are for your benefit.

```

26 // =====
27 class Cat : public Animal{
28 public:
29     virtual ~Cat() {};
30
31     virtual void makeNoise() {};
32
33     // If I do not redine this later, then
34     // this member function is used by default
35     virtual void getTopSpeed(){
36         std::cout << "My top speed is: " << topSpeed << "\n";
37     }
38 protected:
39     int topSpeed; // In mph
40
41 private:
42 };
43
44 // =====
45 // == The Following 3 Classes all inherit from Cat ==
46 // =====
47 class Persian : public Cat{
48 public:
49
50     Persian(){
51         setName("Fluffy");
52         topSpeed = 30;
53     }
54
55     void makeNoise(){
56         std::cout << "meow!\n";
57     }
58
59 private:
60
61 };
62
63 class Tiger : public Cat{
64 public:
65
66     Tiger(){
67         setName("Shere Khan");
68         topSpeed = 45;
69     }
70
71     void makeNoise(){
72         std::cout << "Roar!!\n";
73     }
74
75 private:
76
77 };
78
79 class Cheetah : public Cat{
80 public:
81
82     Cheetah(){

```

```

83     setName("Duma");
84     topSpeed = 70;
85 }
86
87 void makeNoise(){
88     std::cout << "Vroom Vroom\n";
89 }
90
91 private:
92
93 };
94
95
96
97 int main(){
98
99     // Create three different cats
100    // They are each instantiated with different
101    // type, but they all inherit from the same base class
102    Persian meow1;
103    Tiger meow2;
104    Cheetah meow3;
105
106    meow1.getName();
107    meow1.getTopSpeed();
108    meow1.makeNoise();
109    std::cout << '\n';
110
111    meow2.getName();
112    meow2.getTopSpeed();
113    meow2.makeNoise();
114    std::cout << '\n';
115
116    meow3.getName();
117    meow3.getTopSpeed();
118    meow3.makeNoise();
119    std::cout << '\n';
120
121    return 0;
122 }

```

Listing 12.1: Inheritance Example

12.2 Questions

1. Draw a picture, showing the hierarchy of the above code. That is, which classes inherit from the other?
2. Why do we care about inheritance? What does it help us do?
3. Let's say I inherit from a class. Do I automatically 'get everything' (i.e. all the functionality) from the class I inherit from?
4. As an exercise at home, try to split this file into a file that has one class

per file. For the purpose of this activity, it is easier to have everything on one page.

5. What is a virtual member function?
6. What does it mean if I have a function like: `virtual void foo() = 0;` How is this different than a regular virtual member function?