

Part 1 - a, b and c - Video programs

Video Tutorials worth watching

a) Intro to polymorphism Tutorial 55 -

[http://www.youtube.com/watch?v=R\\_PPA9eejDw](http://www.youtube.com/watch?v=R_PPA9eejDw)

b) Virtual Functions -Tutorial 56 (start at 2:19) -

<http://www.youtube.com/watch?v=DudHooleNVg>

c) Abstract Classes and Pure virtual Functions -Tutorial 57 -

<http://www.youtube.com/watch?v=ndz3EHpFEZc>

Write the code and run the program... and Write a summary of these videos:

Part 2 - definition/steps

What is the definition of polymorphism - according to the professor

and

What are the general steps in your code that let use polymorphism

Part 3 - Setup the Virtual function to ENABLE the later selection of the desired function in polymorphism

Create an animal base class.

```
string animalName;
```

```
string sound;
```

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

Create / SETUP 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

mouse

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

Write a switch statement that prints out the animal noise, depending on which animal is chosen.

prompt the user for the type of animal, compare with switch statement, and print out animal sound.

#### Part 4 - Function

(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

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.

#### Part 5 -Compare - Analysis

How is Polymorphism like a Switch statement.

Part 6 - loop

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

Create an animal base class.

```
string animalName;  
  
string sound; // Use constructor to set 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

mouse

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

Declare an array of 4 of type animal

```
animal ** ptrBaseAnimal = new Animal * [4];
```

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

```
example: ptrBaseAnimal[0] = new Cat;
```

do the rest...

Assign the derived address of each to one element in the base array.

Write a for loop to print out the animal noises

```
for ( int i = 0 ; i < 4 ; i++) {  
  
    ptrBaseAnimal[i]->animalSound;
```

