```
#ifndef VirtualPet_h
#define VirtualPet_h
#include <string>
using namespace std;
class VirtualPet
{
   private:
      string name;  //name of the pet
      string color;  //color of the pet
      double baseAdoptionFee;  //base adoption fee for all pet types, default
value $25

   protected:
      string mood;  //mood of the pet - protected makes it easier to program
changeMood
      string type;  //represents the type of Pet for example:  "Dog", "Cat" etc.
      virtual void setType() = 0;  //a virtualPet has a type empty string (see
the constructor) but we want to force derived classes to specify a pet type other
than empty string.  The function has no arguments because we don't want the user to
pass in the type we want the class to choose, so class Cat for example should
override setType and inside just say type = "Cat"
   public:
      VirtualPet(string, string, double = 0);//pass in name, color, base
adoptionFee
      void setName(string);
      string getName()const;

      string getType()const;
      void setColor(string);
      string getColor()const;
      string getMood()const;
      virtual void setMood(string);
      double getBaseAdoptionFee()const;
      void setBaseAdoptionFee(double = 25.0);
      virtual double calcAdoptionCost()const;
      virtual void printInfo()const;  //print's info abou the pet, note the const
      virtual void changeMood();  //randomly changes the pet's mood
      virtual void action()const = 0;  //this funciton will print (cout) a pet
action (like meow) related to a mood, Example let's say the cat's mood is happy the
function might cout << "Meow",note the const
};
#endif /* VirtualPet_h */
```

```
#include "VirtualPet.h"
#include <cstdlib>
#include <ctime>
#include <iostream>
using namespace std;
VirtualPet::VirtualPet(string nme, string clr, double fee)
{
   srand((unsigned int)time(NULL));
   type = "";
   setName(nme);
   setColor(clr);
   setMood("content");
   setBaseAdoptionFee(fee);
}
void VirtualPet::setName(string nm)
{
   name = nm;
}
string VirtualPet::getName()const
{
   return name;
}
string VirtualPet::getType() const
{
   return type;
}
void VirtualPet::setColor(string clr)
{
   color = clr;
}
string VirtualPet::getColor()const
{
   return color;
}
string VirtualPet::getMood()const
{
   return mood;
}
//NOTE:  There are better ways to write this function but keep it simple like this
void VirtualPet::setMood(string md)
{
   if(md == "content" || md == "hungry" || md == "sick")
   {
      mood = md;
```

```
    }
    else
    {
        mood = "content";
    }
}
void VirtualPet::setBaseAdoptionFee(double fee)
{
    if(fee > 0)
    {
        //double base = getBaseAdoptionFee();
        baseAdoptionFee = fee;
    }
    else
    {
        baseAdoptionFee = 25.0;
    }
}
double VirtualPet::getBaseAdoptionFee()const
{
    return baseAdoptionFee;
}
double VirtualPet::calcAdoptionCost()const
{
    return baseAdoptionFee;
}
//to randomly change the virtual pet's mood
void VirtualPet::changeMood()
{
    int num = rand() % 3 + 1;

    switch(num)
    {
        case 1:
            mood = "content";
            break;
        case 2:
            mood = "hungry";
            break;
        case 3:
            mood = "sick";
    }
}
void VirtualPet::printInfo()const
```

```
{
    cout << "\n\nVIRTUAL PET INFO" << endl;
    cout<<  "---------------------" << endl;
    cout << "Adoption Cost: "<< calcAdoptionCost() << endl;
    cout << "Name:          "<< getName() << endl;
    cout << "Type:          "<< getType() << endl;
    cout << "Color:         "<< getColor() << endl;
    cout << "Mood:          " << getMood() << endl;

}
```