|  |  |  |
| --- | --- | --- |
| American University of SharjahSchool of Engineering Computer Engineering Department  P. O. Box 26666 Sharjah, UAE |  | Lab Instructor: Eng. Sameer Alawnah  **Office**: EB2 - 101  **Phone**: 06-515 2974  **e-mail**: salawnah@aus.edu  **Semester**: Fall 2016 |

**CMP 220L – Introduction to Computer Science II**

**Inheritance**

**LAB Assignment 10**

**Exercise 1**

An **animal** has a **height** and a **weight**. A **herbivore** is an **animal** and has an additional feature that it eatssome vegetarian food. The default values for height and weight of an animal are 3.5 ft and 50 kg respectively. The default values for height and weight of a herbivore are 4.0 ft and 75 kg respectively.

1. Define and implement the classes **Animal** and **Herbivore** assuming that the **attributes** are **private** in both the classes. (The eat attribute in herbivore must be of type **string**)
2. Provide default constructor, regular constructor and copy constructor (if needed).
3. Provide set and get functions for the private data in both the classes.
4. Provide a member function for the assignment (=) operator (if needed .
5. Provide a print function in each class to print attribute details.
6. Provide a destructor in each class (if needed)
7. Provide a driver to test all the member functions
8. Note: We know that you can implement the herbivore functions without calling any of the Animal getters/getters, BUT you have to use them in either the constructor or the printDetails functions).

**Submissions**: Animal.h, Animal.cpp, Herbivore.h and Herbivore.cpp

Sample Main program:

#include "Herbivore.h"

#include <iostream>

using namespace std;

void main()

{

Animal a1, a2(5, 100), a3(a2), a4 = a1;

Herbivore h1, h2(6, 150, "fruits"), h3(h2), h4 = h1;

a1.printDetails();

cout << endl;

a2.printDetails();

cout << endl;

a3.printDetails();

cout << endl;

a4.printDetails();

cout << endl;

h1.printDetails();

cout << endl;

h2.printDetails();

cout << endl;

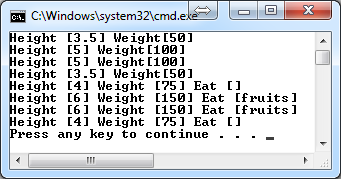
h3.printDetails();

cout << endl;

h4.printDetails();

cout << endl;

}



#include<iostream>

#ifndef ANIMAL\_H

#define ANIMAL\_H

using namespace std;

class Animal

{

private:

double height;

double weight;

public:

void setHeight(double h);

void setWeight(double w);

double get\_height();

double get\_weight();

Animal();

Animal(double newheight, double newweight);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include"Animal.h"

#include<iostream>

void Animal::setHeight(double h)

{

height = h;

}

void Animal::setWeight(double w)

{

weight = w;

}

double Animal::get\_height()

{

return height;

}

double Animal::get\_weight()

{

return weight;

}

Animal::Animal()

{

weight = 50;

height = 3.5;

}

Animal::Animal(double newheight, double newweight)

{

height = newheight;

weight = newweight;

}

void Animal::printDetails()

{

cout << "Height [" << height << "] Weight [" << weight << "]" << endl;

}

**\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

#ifndef HERIVORE\_H

#define HERBIVORE\_H

#include"Animal.h"

#include <string>

using namespace std;

class Herbivore : public Animal

{

private:

double height;

double weight;

string eat;

public:

void setHeight(double h);

void setWeight(double w);

void setEat(string e);

double get\_height();

double get\_weight();

string get\_eat();

Herbivore();

Herbivore(double newheight, double newweight, string neweat);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include"Herbivore.h"

void Herbivore::setHeight(double h)

{

height = h;

}

void Herbivore::setWeight(double w)

{

weight = w;

}

void Herbivore::setEat(string e)

{

eat = e;

}

double Herbivore::get\_height()

{

return height;

}

double Herbivore::get\_weight()

{

return weight;

}

string Herbivore::get\_eat()

{

return eat;

}

Herbivore::Herbivore()

{

height = 4;

weight = 75;

}

Herbivore::Herbivore(double newheight, double newweight, string neweat)

{

height = newheight;

weight = newweight;

eat = neweat;

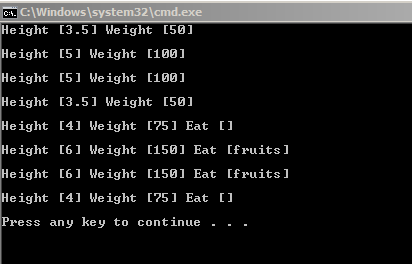
}

void Herbivore::printDetails()

{

cout << "Height [" << get\_height() << "] Weight [" << get\_weight() << "] Eat [" << eat << "]" << endl;

}



**Exercise 2**

Make **attributes** in the parent class (animal) as **protected** and modify the implementation for the **herbivore** class and test it in the same way as in **exercise 1.**

**Hint: Direct accessibility** of the **protected** members in a parent class is **extended** to the sub classes.

**Submissions:** Animal.h, Herbivore.cpp

#include<iostream>

#ifndef ANIMAL\_H

#define ANIMAL\_H

using namespace std;

class Animal

{

protected:

double height;

double weight;

public:

Animal();

Animal(double newheight, double newweight);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include"Herbivore.h"

Herbivore::Herbivore()

{

height = 4;

weight = 75;

}

Herbivore::Herbivore(double newheight, double newweight, string neweat)

{

height = newheight;

weight = newweight;

eat = neweat;

}

void Herbivore::printDetails()

{

cout << "Height [" << height << "] Weight [" << weight << "] Eat [" << eat << "]" << endl;

}

**Exercise 3** **(built on the top of Exercise 1)**

A **PetHerbiAnimal** is a **herbivore** and has an additional **feature** that it has a **name.**

The default values for height and weight of a herbivore are 2.5 ft and 30 kg respectively.

1. Define the class **PetHerbiAnimal** and implement it.
2. Provide default constructor, regular constructor and copy constructor (if needed)
3. Provide set and get functions for the private in both the classes
4. Provide a member function for the assignment(=) operator ( if needed)
5. Provide a destructor for each class ( if needed)
6. Provide a driver to test all the member functions**.**
7. **name** attribute must beof type **string**
8. Note: We know that you can implement the herbivore functions without calling any of the Herbivore getters/getters, BUT you have to use them in either the constructor or the printDetails functions).

**Note:** The attributes (member variables) must be **private** in each class in the inheritance hierarchy.

**Submissions:** Animal.h, Animal.cpp, Herbivore.h, Herbivore.cpp, PetHerbiAnimal.h, PetHerbiAnimal.cpp

**Main program:**

#include "PetHerbiAnimal.h"

#include<iostream>

using namespace std;

void main()

{

PetHerbiAnimal p1, p2(0.5, 3, "leaves", "rabbit"), p3(p2), p4 = p1;

p1.printDetails();

cout << endl;

p2.printDetails();

cout << endl;

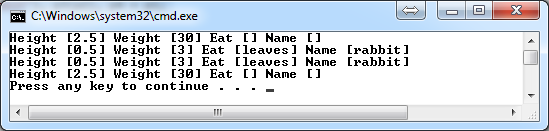
p3.printDetails();

cout << endl;

p4.printDetails();

cout << endl;

}



#include<iostream>

#ifndef ANIMAL\_H

#define ANIMAL\_H

using namespace std;

class Animal

{

private:

double height;

double weight;

public:

void setHeight(double h);

void setWeight(double w);

double get\_height();

double get\_weight();

Animal();

Animal(double newheight, double newweight);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\

#ifndef HERIVORE\_H

#define HERBIVORE\_H

#include"Animal.h"

#include <string>

using namespace std;

class Herbivore : public Animal

{

private:

double height;

double weight;

string eat;

public:

void setHeight(double h);

void setWeight(double w);

void setEat(string e);

double get\_height();

double get\_weight();

string get\_eat();

Herbivore();

Herbivore(double newheight, double newweight, string neweat);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include "Herbivore.h"

#include <string>

#ifndef PETHERBIANIMAL\_H

#define PETHERBIANIMAL\_H

class PetHerbiAnimal : public Herbivore

{

private:

double height;

double weight;

string eat;

string name;

public:

void setHeight(double h);

void setWeight(double w);

void setEat(string e);

void setName(string n);

double get\_height();

double get\_weight();

string get\_eat();

string get\_name();

PetHerbiAnimal();

PetHerbiAnimal(double newheight, double newweight, string neweat, string newname);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include"Animal.h"

#include<iostream>

void Animal::setHeight(double h)

{

height = h;

}

void Animal::setWeight(double w)

{

weight = w;

}

double Animal::get\_height()

{

return height;

}

double Animal::get\_weight()

{

return weight;

}

Animal::Animal()

{

weight = 50;

height = 3.5;

}

Animal::Animal(double newheight, double newweight)

{

height = newheight;

weight = newweight;

}

void Animal::printDetails()

{

cout << "Height [" << height << "] Weight [" << weight << "]" << endl;

}

**\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

#ifndef HERIVORE\_H

#define HERBIVORE\_H

#include"Animal.h"

#include <string>

using namespace std;

class Herbivore : public Animal

{

private:

double height;

double weight;

string eat;

public:

void setHeight(double h);

void setWeight(double w);

void setEat(string e);

double get\_height();

double get\_weight();

string get\_eat();

Herbivore();

Herbivore(double newheight, double newweight, string neweat);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include "Herbivore.h"

#include <string>

#ifndef PETHERBIANIMAL\_H

#define PETHERBIANIMAL\_H

class PetHerbiAnimal : public Herbivore

{

private:

double height;

double weight;

string eat;

string name;

public:

void setHeight(double h);

void setWeight(double w);

void setEat(string e);

void setName(string n);

double get\_height();

double get\_weight();

string get\_eat();

string get\_name();

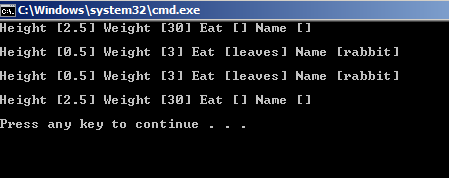
PetHerbiAnimal();

PetHerbiAnimal(double newheight, double newweight, string neweat, string newname);

void printDetails();

};

#endif



**Exercise 4 (built on the top of Exercise 2)**

Change the access type of the attributes from **private** to **protected** in the **Animal** class and the **Herbivore** class; modify the implementation for each class, also modify the PetHerbiAnimal to make use of the new access level and test them in the same way as in **exercise 3**

**Submissions:** Herbivore.h and PetHerbiAnimal.cpp

#ifndef HERIVORE\_H

#define HERBIVORE\_H

#include"Animal.h"

#include <string>

using namespace std;

class Herbivore : public Animal

{

protected:

double height;

double weight;

string eat;

public:

void setEat(string e);

Herbivore();

Herbivore(double newheight, double newweight, string neweat);

void printDetails();

};

#endif

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#include "PetHerbiAnimal.h"

PetHerbiAnimal::PetHerbiAnimal()

{

height = 2.5;

weight = 30;

}

PetHerbiAnimal::PetHerbiAnimal(double newheight, double newweight, string neweat, string newname)

{

height = newheight;

weight = newweight;

eat = neweat;

name = newname;

}

void PetHerbiAnimal::printDetails()

{

cout << "Height [" << height << "] Weight [" << weight << "] Eat [" << eat << "] Name ["<<name<<"] " << endl;

}

Good Luck ☺