|  |  |  |
| --- | --- | --- |
| **American University of Sharjah**  **College of Engineering**  Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE | A picture containing logo  Description automatically generated | **Instructors:** Dr. Aliaa Moualla  **Lab Instructor:** Sameer Alawnah  **Office:** EB1-0012C  **Phone**: 971-6-515-4940  **e-mail**: salawnah@aus.edu  **Semester**: Spring 2024 |

**CMP 220L - Programming II**

**Lab #10 – Inheritance**

**Note: using ChatGPT will be considered a violation of the AUS integrity code.**

**Objectives:**

* Practice building classes by using inheritance.

Using Visual Studio 2022, write the below programs, compile and provide screenshots of output.

Note: you are required to submit copy of the code + screenshots of program run for each exercise.

A cat has a name, weight, and age. A cat can be pet. You can also give food to a cat, and they will eat it. A Cheshire cat is a special kind of cat with magical abilities. In addition to having a name, weight, and age, a Cheshire cat can be visible or invisible. A Cheshire cat cannot be pet but will eat food if offered only between 8am and 8pm.

Write classes to model the cat and the Cheshire cat.

**Exercise #1**

Create a **Cat** class which has the following attributes:

* + **Name**
  + **Age**
  + **Weight**

and the following functions:

* Default constructor.
* Parametrized constructor.
* *Feed* which takes **food (string)** as a parameter and prints what the cat is eating.
* *Pet* which prints that the cat is being pet.
* *display* which prints the cat’s information.

#include <iostream>

#include <string>

**using** **namespace** std;

**class** Cat {

**private**:

string name;

**int** age;

**double** weight;

**public**:

Cat() {

name = "Unnamed";

age = 0;

weight = 0.0;

}

Cat(**int** a, string n, **double** w) {

name = n;

age = a;

weight = w;

}

**void** feed(string food) {

cout << name << " is eating " << food << "." << endl;

}

**void** pet() {

cout << "You pet " << name << "." << endl;

}

**void** display() {

cout << "Name: " << name << endl;

cout << "Age: " << age << " years" << endl;

cout << "Weight: " << weight << " kg" << endl;

}

};

**int** main() {

Cat myCat(5, "Whiskers", 4.2);

//Cheshire cheshireCat(50,"Cheshire", 3.5);

myCat.feed("Fish");

myCat.pet();

myCat.display();

cout << endl;

//cheshireCat.appear();

//cheshireCat.display();

//cheshireCat.feed("glitter", 14);// Cheshire cat should eat because it's between 8am and 8pm

//cheshireCat.feed("sprinkles", 21);

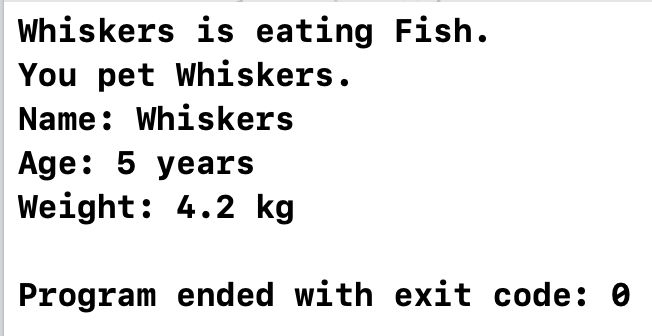
//cheshireCat.pet(); // Cheshire cat does not like being pet

//cheshireCat.disappear(); // Cheshire cat disappears

//cheshireCat.display();

**return** 0;

}

****

**Exercise 2**

Create a class **Cheshire** which is a child of **Cat** and has the following additional attributes:

* + **Visible** which can be set to true or false.

and the following functions:

* Default constructor which calls the parent’s constructor and sets Visible to false.
* Parametrized constructor which calls the parent’s constructor and sets Visible to false.
* Overloaded *Feed* which takes both the **food (string)** and **time (int)** as parameters. If the time is within the Cheshire cat’s mealtime (between 8am and 8pm), it should call the normal Feed. Else, it should print that the Cheshire cat is not in the mood to eat. Assume 24 hour format for the time.
* Redefined *Pet* which prints that the Cheshire cat does not like being pet.
* Redefined *display* which calls the parent’s Print function, then prints whether the cat is visible or not.
* *Appear* and *Disappear* which change the state of the Visible attribute

Use the following **main** program to test your classes

int main() {

Cat myCat(5, "Whiskers", 4.2);

Cheshire cheshireCat(50,"Cheshire", 3.5);

myCat.feed("Fish");

myCat.pet();

myCat.display();

cout << endl;

cheshireCat.appear();

cheshireCat.display();

cheshireCat.feed("glitter", 14);// Cheshire cat should eat because it's between 8am and 8pm

cheshireCat.feed("sprinkles", 21);

cheshireCat.pet(); // Cheshire cat does not like being pet

cheshireCat.disappear(); // Cheshire cat disappears

cheshireCat.display();

return 0;

}

A screenshot of a computer

Description automatically generated**Sample output**

#include <iostream>

#include <string>

**using** **namespace** std;

**class** Cat {

**protected**:

string name;

**int** age;

**double** weight;

**public**:

Cat() : name(""), age(0), weight(0.0) {}

Cat(**int** age, string name, **double** weight) : name(name), age(age), weight(weight) {}

**void** feed(string food) {

cout << name << " is eating " << food << endl;

}

**void** pet() {

cout << name << " is enjoying being pet" << endl;

}

**void** display() {

cout << name << " is " << age << " years old and weighs " << weight << endl;

}

};

**class** Cheshire : **public** Cat {

**private**:

**bool** visible;

**public**:

Cheshire() : Cat(), visible(**false**) {}

Cheshire(**int** age, string name, **double** weight) : Cat(age, name, weight), visible(**false**) {}

**void** feed(string food, **int** time) {

**if** (time >= 8 && time <= 20) {

Cat::feed(food);

} **else** {

cout << name << " is not in the mood to eat" << endl;

}

}

**void** pet() {

cout << name << " doesn't like being pet" << endl;

}

**void** display() {

Cat::display();

cout << "he is currently " << (visible ? "visible" : "invisible") << endl;

}

**void** appear() {

visible = **true**;

}

**void** disappear() {

visible = **false**;

}

};

**int** main() {

Cat myCat(5, "Whiskers", 4.2);

Cheshire cheshireCat(50, "Cheshire", 3.5);

myCat.feed("Fish");

myCat.pet();

myCat.display();

cout << endl;

cheshireCat.appear();

cheshireCat.display();

cheshireCat.feed("glitter", 14); // Cheshire cat should eat because it's between 8am and 8pm

cheshireCat.feed("sprinkles", 21);

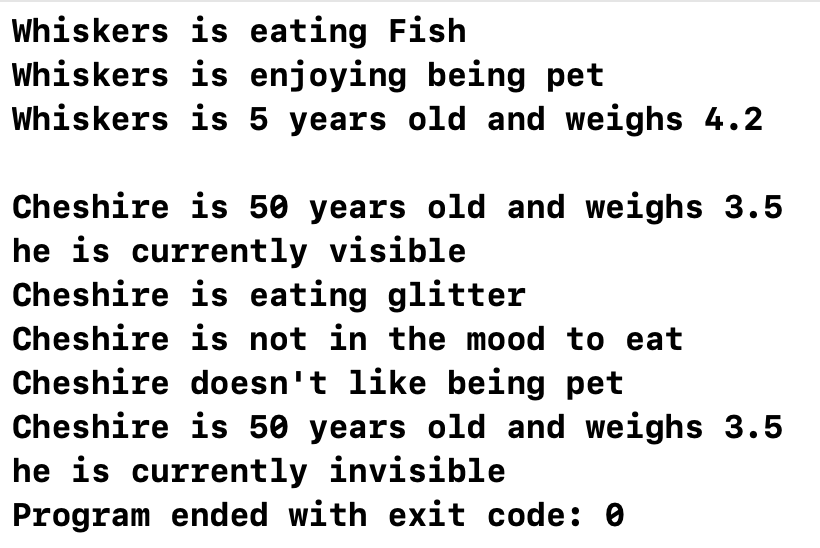
cheshireCat.pet(); // Cheshire cat does not like being pet

cheshireCat.disappear(); // Cheshire cat disappears

cheshireCat.display();

**return** 0;

}

****