|  |  |  |
| --- | --- | --- |
| **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 – Polymorphism**

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

**Objectives:**

* Practice building classes by using inheritance.
* Practice using polymorphism.

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.

**Exercise #1**

An art gallery contains a collection of Art Pieces which can be categorized into Paintings and Sculptures. Create classes to model the art gallery.

1. Create a class for **Artpiece** which has name (string), artist (string), and the date it was created (int) as data members.
2. Create a class for **Painting** which is a child of **Artpiece** and additionally has type of paint (string), height(double), and width(double) as data members.
3. Create a class for **Sculpture** which is a child of **Artpiece** and additionally has weight (double) as a data member.

For each of the classes, define:

* Default constructor
* Parametrized constructor
* Copy constructor
* Overloaded assignment operator
* Destructor
* Print function which outputs the class’s data members

Outside the classes, define the following function:

* *Evaluate* which takes a reference to an Artpiece instance as input, and outputs the name of art piece, the artist, and the date, as well as a random price between $1 and $100,000,000.

Notes (make sure to replace {Class Name} with the actual class name:

* In every Default constructor, you should print: “Default Const of {Class Name} is called“
* In every Parametrized constructor, you should print: “Parametrized Const of {Class Name} is called“
* In every Copy constructor, you should print: “Copy Const of {Class Name} is called“
* In every Destructor, you should print: “Destructor of {Class Name} is called“
* In every = operator, you should print: “= operator for {Class Name} is called “
* Since we don’t have any dynamic memory, your classes will work fine without the Big-3, but we include them to show you the order of execution of them.

Use the following main to test your classes:

int main() {

srand(time(NULL));

Painting p1("The Potato Eaters", "Vincent Van Gogh", 1885, "Oil", 0.82, 1.14);

cout << "First painting created using the parametrized constructor" << endl;

p1.print();

cout << endl;

Painting p2 = p1;

cout << "Second painting created using the parametrized constructor" << endl;

p2.print();

cout << endl;

Sculpture s1("Guitar", "Pablo Picasso", 1912, 54.2);

cout << "First sculpture created using the parametrized constructor" << endl;

s1.print();

cout << endl;

Sculpture s2;

cout << "Second sculpture created using the default constructor" << endl;

s2 = s1;

cout << "Second sculpture after the assignment operator" << endl;

s2.print();

cout << endl;

cout << "Evaluating the first painting" << endl;

Evaluate(p1);

cout << endl;

cout << "Evaluating the first sculpture" << endl;

Evaluate(s1);

cout << endl;

return 0;

}

Sample Output:

Parametrized Const of Artpiece is called

Parametrized Const of Painting is called

First painting created using the parametrized constructor

Name: The Potato Eaters

Artist: Vincent Van Gogh

Creation Date: 1885

Paint Type: Oil

Painting dimensions 1.14 x 0.82m

Copy Const of Artpiece is called

Copy Const of Painting is called

Second painting created using the parametrized constructor

Name: The Potato Eaters

Artist: Vincent Van Gogh

Creation Date: 1885

Paint Type: Oil

Painting dimensions 1.14 x 0.82m

Parametrized Const of Artpiece is called

Parametrized Const of Sculpture is called

First sculpture created using the parametrized constructor

Name: Guitar

Artist: Pablo Picasso

Creation Date: 1912

Weight: 54.2kg

Default Const of Artpiece is called

Default Const of Sculpture is called

Second sculpture created using the default constructor

= operator for Artpiece is called

= operator for Sculpture is called

Second sculpture after the assignment operator

Name: Guitar

Artist: Pablo Picasso

Creation Date: 1912

Weight: 54.2kg

Evaluating the first painting

Name: The Potato Eaters

Artist: Vincent Van Gogh

Creation Date: 1885

The value of this artpiece: $80534003

Evaluating the first sculpture

Name: Guitar

Artist: Pablo Picasso

Creation Date: 1912

The value of this artpiece: $16350219

Destructor of Sculpture of is called

Destructor of Artpiece of is called

Destructor of Sculpture of is called

Destructor of Artpiece of is called

Destructor of Painting of is called

Destructor of Artpiece of is called

Destructor of Painting of is called

Destructor of Artpiece of is called

#include <iostream>

#include<string>

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

**class** artpiece {

**private**:

string name, artist;

**int** date;

**public**:

artpiece(){ cout << "Default Const of artpiece is called" << endl; }

artpiece(string n, string a, **int** d) : name(n), artist(a), date(d) { cout << "Parametrized Const artpice is called" << endl; }

artpiece(artpiece& obj) : name(obj.name), artist(obj.artist), date(obj.date) { cout << "Copy Const of artpiece is called" << endl; }

artpiece& **operator**=(**const** artpiece& obj) {

name = obj.name;

artist = obj.artist;

date = obj.date;

cout << "= operator for artpiece is called “" << endl;

**return**\***this**;

}

**void** print() {

cout << "name: " << name << " artist: " << artist << " date: " << date;

}

~artpiece(){ cout << "Destructor of artpiece is called" << endl; };

};

**class** painting: **public** artpiece {

**private**:

string paint;

**float** height, width;

**public**:

painting() { cout << "Default Const of Painting is called" << endl; }

painting(string n, string a, **int** d, string p, **float** h, **float** w): artpiece(n,a,d), paint(p), height(h), width(w) { cout << "Parametrized Const painting is called" << endl; }

painting(painting& obj) :artpiece(obj), height(obj.height), width(obj.width) { cout << "Copy Const of painting is called" << endl; }

painting& **operator** =(**const** painting& obj) {

artpiece::**operator**=(obj);

paint = obj.paint;

height = obj.height;

width = obj.width;

cout << "= operator for painting is called “" << endl;

**return**\***this**;

}

**void** print() {

artpiece::print();

cout<<" paint: "<<paint<<" heiht: "<<height<<" width: "<<width;

}

~painting() { cout << "Destructor of painting is called" << endl; };

};

**class** sculpture : **public** artpiece {

**private**:

**float** weight;

**public**:

sculpture() { cout << "Default Const of sculpture is called" << endl; }

sculpture(string n, string a, **int** d, **float** wt) : artpiece(n, a, d), weight(wt) { cout << "Parametrized Const sculpture is called" << endl; }

sculpture(sculpture& obj) :artpiece(obj), weight(obj.weight) { cout << "Copy Const of sculpture is called" << endl; }

sculpture& **operator** =(**const** sculpture& obj) {

artpiece::**operator**=(obj);

weight = obj.weight;

cout << "= operator for sculpture is called “" << endl;

**return**\***this**;

}

**void** print() {

artpiece::print();

cout << "weight: " << weight;

}

~sculpture() { cout << "Destructor of sculpture is called" << endl; };

};

**void** evaluate(artpiece& obj) {

obj.print();

**int** random\_price = rand() % 100000000 + 1;

cout << "The value of this artpiece: " << random\_price << endl;

}

**int** main()

{

srand(time(**NULL**));

painting p1("The Potato Eaters", "Vincent Van Gogh", 1885, "Oil", 0.82, 1.14);

cout << "First painting created using the parametrized constructor" << endl;

p1.print();

cout << endl;

painting p2 = p1;

cout << "Second painting created using the parametrized constructor" << endl;

p2.print();

cout << endl;

sculpture s1("Guitar", "Pablo Picasso", 1912, 54.2);

cout << "First sculpture created using the parametrized constructor" << endl;

s1.print();

cout << endl;

sculpture s2;

cout << "Second sculpture created using the default constructor" << endl;

s2 = s1;

cout << "Second sculpture after the assignment operator" << endl;

s2.print();

cout << endl;

cout << "Evaluating the first painting" << endl;

evaluate(p1);

cout << endl;

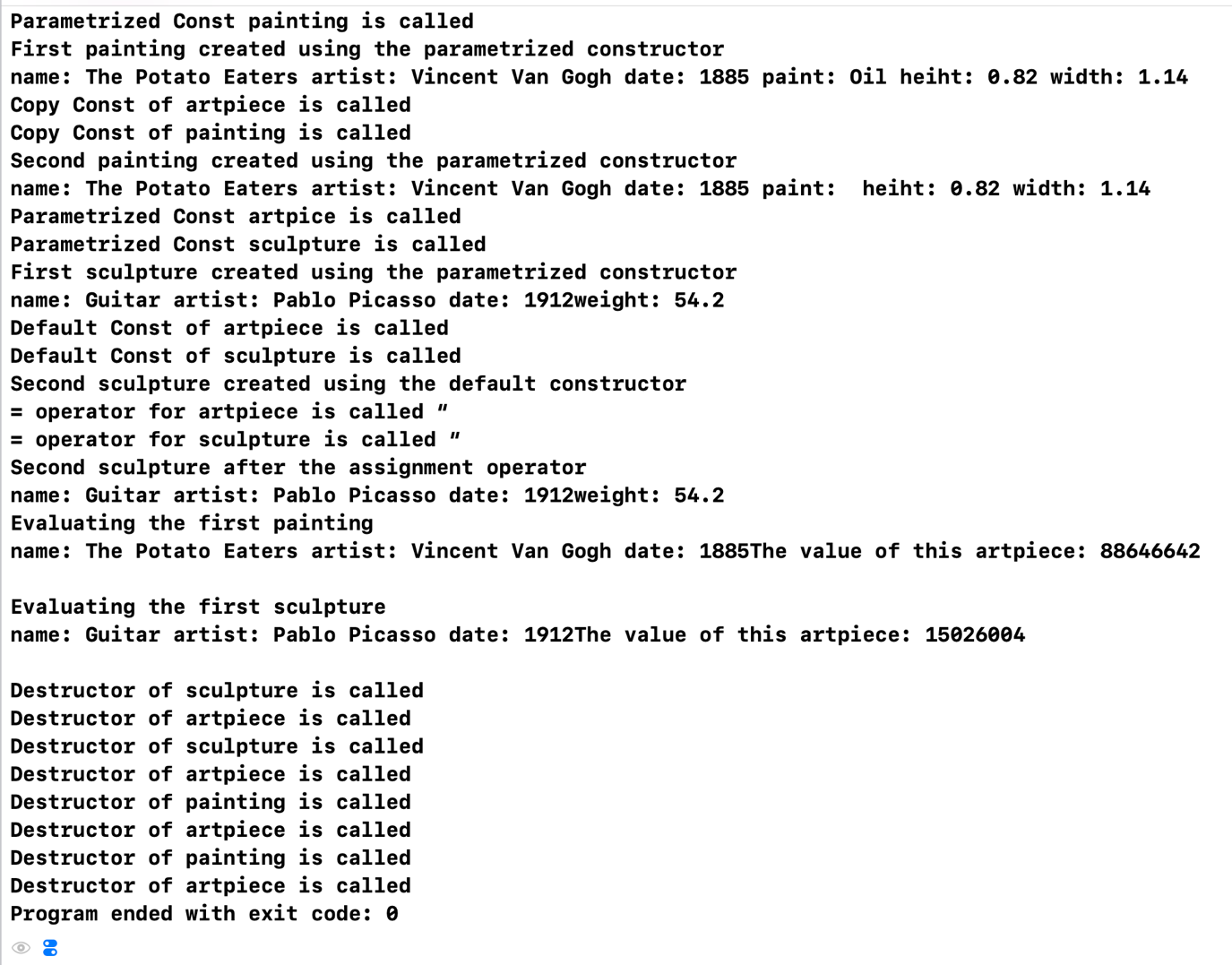
cout << "Evaluating the first sculpture" << endl;

evaluate(s1);

cout << endl;

**return** 0;

}



**Exercise 2**

Before a museum receives a new collection of art pieces, they must calculate the cubic meter space needed to display them. The calculation goes as follows:

* Paintings are hanged on the wall and only need around double their area on the wall and a 2m space in front of the wall, so people are not too close, therefore spaces needed is 2 x height x width x 2m.
* Sculptures are always reserved a space of 1m x 1m x 4m x (0.5 \* weight).

Declare EstimateSpace as a pure virtual function in the Artpiece class, then define it in the child classes to return the appropriate space calculation.

Make sure to make the print function a virtual function in the Aritpiece class, also the Destructor should be virtual.

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

int main() {

srand(time(NULL));

Painting p1("The Potato Eaters", "Vincent Van Gogh", 1885, "Oil", 0.82, 1.14);

Painting p2 = p1;

Sculpture s1("Guitar", "Pablo Picasso", 1912, 54.2);

Sculpture s2;

Artpiece\* collection[3];

collection[0] = &s1;

collection[1] = &p1;

collection[2] = new Painting("Composition VIII", "Wassily Kandinsky", 1923, "Oil", 1.04, 2.01);

cout << "The museum received three new artpieces as follows:" << endl;

for (int i = 0; i < 3; i++) {

collection[i]->print();

cout << endl;

}

cout << "Total cubic meter space needed to display the artpieces: ";

double space = 0;

for (int i = 0; i < 3; i++) {

space += collection[i]->EstimateSpace();

}

cout << space << "m^3"<<endl;

delete collection[2];

return 0;

}

Sample output:

Parametrized Const of Artpiece is called

Parametrized Const of Painting is called

Copy Const of Artpiece is called

Copy Const of Painting is called

Parametrized Const of Artpiece is called

Parametrized Const of Sculpture is called

Default Const of Artpiece is called

Default Const of Sculpture is called

Parametrized Const of Artpiece is called

Parametrized Const of Painting is called

The museum received three new artpieces as follows:

Name: Guitar

Artist: Pablo Picasso

Creation Date: 1912

Weight: 54.2kg

Name: The Potato Eaters

Artist: Vincent Van Gogh

Creation Date: 1885

Paint Type: Oil

Painting dimensions 1.14 x 0.82m

Name: Composition VIII

Artist: Wassily Kandinsky

Creation Date: 1923

Paint Type: Oil

Painting dimensions 2.01 x 1.04m

Total cubic meter space needed to display the artpieces: 120.501m^3

Destructor of Painting of is called

Destructor of Artpiece of is called

Destructor of Sculpture of is called

Destructor of Artpiece of is called

Destructor of Sculpture of is called

Destructor of Artpiece of is called

Destructor of Painting of is called

Destructor of Artpiece of is called

Destructor of Painting of is called

Destructor of Artpiece of is called

#include <iostream>

#include<string>

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

**class** artpiece {

**private**:

string name, artist;

**int** date;

**public**:

artpiece() { cout << "Default Const of artpiece is called" << endl; }

artpiece(string n, string a, **int** d) : name(n), artist(a), date(d) { cout << "Parametrized Const artpice is called" << endl; }

artpiece(artpiece& obj) : name(obj.name), artist(obj.artist), date(obj.date) { cout << "Copy Const of artpiece is called" << endl; }

artpiece& **operator**=(**const** artpiece& obj) {

name = obj.name;

artist = obj.artist;

date = obj.date;

cout << "= operator for artpiece is called “" << endl;

**return**\***this**;

}

**virtual** **void** print() {

cout << "name: " << name << " artist: " << artist << " date: " << date;

}

**virtual** ~artpiece() { cout << "Destructor of artpiece is called" << endl; };

**virtual** **int** estimate()=0;

};

**class** painting : **public** artpiece {

**private**:

string paint;

**float** height, width;

**public**:

painting() :artpiece() { cout << "Default Const of Painting is called" << endl; }

painting(string n, string a, **int** d, string p, **float** h, **float** w) : artpiece(n, a, d), paint(p), height(h), width(w) { cout << "Parametrized Const painting is called" << endl; }

painting(painting& obj) :artpiece(obj), height(obj.height), width(obj.width) { cout << "Copy Const of painting is called" << endl; }

painting& **operator** =(**const** painting& obj) {

artpiece::**operator**=(obj);

paint = obj.paint;

height = obj.height;

width = obj.width;

cout << "= operator for painting is called “" << endl;

**return**\***this**;

}

**void** print() {

artpiece::print();

cout << " paint: " << paint << " heiht: " << height << " width: " << width;

}

**virtual** **int** estimate() {

**return**(4 \* height \* width);

}

~painting() { cout << "Destructor of painting is called" << endl; };

};

**class** sculpture : **public** artpiece {

**private**:

**float** weight;

**public**:

sculpture() :artpiece() { cout << "Default Const of sculpture is called" << endl; }

sculpture(string n, string a, **int** d, **float** wt) : artpiece(n, a, d), weight(wt) { cout << "Parametrized Const sculpture is called" << endl; }

sculpture(sculpture& obj) :artpiece(obj), weight(obj.weight) { cout << "Copy Const of sculpture is called" << endl; }

sculpture& **operator** =(**const** sculpture& obj) {

artpiece::**operator**=(obj);

weight = obj.weight;

cout << "= operator for sculpture is called “" << endl;

**return**\***this**;

}

**void** print() {

artpiece::print();

cout << "weight: " << weight;

}

**virtual** **int** estimate() {

**return**(4 \* 0.5 \* weight);

}

~sculpture() { cout << "Destructor of sculpture is called" << endl; };

};

**void** evaluate(artpiece& obj) {

obj.print();

**int** random\_price = rand() % 100000000 + 1;

cout << "The value of this artpiece: " << random\_price << endl;

}

**int** main() {

srand(time(**NULL**));

painting p1("The Potato Eaters", "Vincent Van Gogh", 1885, "Oil", 0.82, 1.14);

painting p2 = p1;

sculpture s1("Guitar", "Pablo Picasso", 1912, 54.2);

sculpture s2;

artpiece\* collection[3];

collection[0] = &s1;

collection[1] = &p1;

collection[2] = **new** painting("Composition VIII", "Wassily Kandinsky", 1923, "Oil", 1.04, 2.01);

cout << "The museum received three new artpieces as follows:" << endl;

**for** (**int** i = 0; i < 3; i++) {

collection[i]->print();

cout << endl;

}

cout << "Total cubic meter space needed to display the artpieces: ";

**double** space = 0;

**for** (**int** i = 0; i < 3; i++) {

space += collection[i]->estimate();

}

cout << space << "m^3" << endl;

**delete** collection[2];

**return** 0;

}

A screenshot of a computer

Description automatically generated