OBJECT ORIENTED PROGRAMMING LAB

Lab Manual # 07

Composition in C++

Instructor: Mazhar Iqbal

Semester Spring, 2023

Course Code: CL1004

Fast National University of Computer and Emerging Sciences
Peshawar

Department of Computer Science

OBJECT ORIENTED PROGRAMMING LANGUAGE

Table of Contents

Static Variable

In C++, static is a keyword or modifier that belongs to the type not instance. So instance is not required to access the static members.

Memory efficient: Now we don't need to create instance for accessing the static members, so it saves memory.

Moreover, it belongs to the type, so it will not get memory each time when instance is created.

C++ Static Field

- A field which is declared as static is called static field.
- Unlike instance field which gets memory each time whenever you create object, there is only one copy of static field created in the memory.
- It is shared to all the objects.
- It is used to refer the common property of all objects such as rateOfInterest in case of Account, companyName in case of Employee etc.

Let's see the simple example of static field in C++.

C++ static members example

In road-racing game, for example, a race car might want to know how many other cars are still in the race. In this case a static variable count could be included as a member of the class. All the objects would have access to this variable. It would be the same variable. It would be the same variable for all of them; they would all see the same count.

Example Code demonstrating Static Variable and Static Method

```
#include<iostream>
using namespace std;
class Account {
  public:
     int accno; //data member (also instance variable)
     string name;
     static int count;
     Account(int accno, string name)
     {
        this->accno = accno;
        this->name = name;
        count++;
     }
     void display()
     {
        cout<<"\nAccount no: "<<accno<<"\nAcc Name: "<<name<<endl;
     }
     static void displayCount()
     {
          cout<<"\nTotal Objects: "<<count<<endl;
     }
};
int Account::count=0;
int main() {

   Account a1 =Account(201, "Ali"); //creating an object of Account

   Account a2=Account(202, "Saad");
   Account a3=Account(203, "Sharjeel");
   a1.display();
   a2.display();
   a3.display();
   a1.displayCount();
   Account::displayCount();
   return 0;
}
```

Composition

In object-oriented programming languages, object composition is used for objects that have a **"has-a"** relationship with each other. **For example**, a mobile has-a battery, has-a sensor, has-a

screen, etc. Therefore, the complex object is called the whole or a parent object whereas a simpler object is often referred to as a child object. In this case, all the objects or components are the child objects which together make up the complex object(mobile).

A class can have one or more objects of other classes as members. A class is written in such a way that the object of another existing class becomes a member of the new class. this relationship between classes is known as C++ Composition. It is also known as containment, part-whole, or has-a relationship. A common form of software reusability is C++ Composition.

In C++ Composition, an object is a part of another object. The object that is a part of another object is known as a sub-object. When a C++ Composition is destroyed, then all of its subobjects are destroyed as well. Such as when a car is destroyed, then its motor, frame, and other parts are also destroyed with it. It has a do and die relationship.

Classes that have data members of built-in type work really well for simple classes. However, in real-world programming, the product or software comprises of many different smaller objects and classes. In the above example, the mobile phone consisted of various different objects that on a whole made up a complete mobile phone. Since each of those objects performs a different task, they all are maintained in different classes. Therefore, this concept of complex objects is being used in most of the real-world scenarios. The advantages of using this concept are:

The most important advantage is if any changes have to be made in a child class, only the child class can be changed rather than changing the entire parent class.

- For example, if we want to change the battery class in the mobile object, with the help of composition, the changes are only made in the battery class and the entire mobile object works fine with the updated changes.

Example Code

```
// C++ program to implement a composition

// A point class
class Point {
private:
      int x;
      int y;
};

class Location {
      Private:
      Point Source;
      Point Destination
};
```

In the classes given here, **Location** uses objects of class Point as its data members. Hence, Location is a complex class which uses a simple class Point. Let us have a look at the program that makes use of the composition.

Program Example

```
#include<iostream>
using namespace std;
class DOB
{
	int  day;
	int month;
	int year;
	public:
		int getDay()
		{
			return day;
		}
		int getMonth()
		{
			return month;
		}
		int getYear()
		{
			return year;
		}
		void setDay(int day)
		{
			this->day=day;
		}
		void setMonth(int month)
		{
			this->month=month;

		}
		void setYear(int year)
		{
			this->year=year;
		}
};
```

```
class Person
{
	string name;
	DOB date;
	public:
		Person(string name, int day, int month, int year)
		{
			this->name=name;
			this->date.setDay(day);
			this->date.setMonth(month);
			this->date.setYear(year);
		}
		void setDate(int day, int month, int year)
		{
			this->date.setDay(day);
			this->date.setMonth(month);
			this->date.setYear(year);
		}
		void setName(string name)
		{
			this->name=name;
		}
		string getDate()
		{
			return to_string(date.getDay()) + "/" + to_string(date.getMonth()) + "/" +
			to_string(date.getYear());

		}
		string getName()
		{
			return name;
		}
};
```

```
int main()
{
        Person p("khan", 28, 2, 1999);
        cout<<p.getName()<<endl;
        cout<<p.getDate()<<endl;
}
```

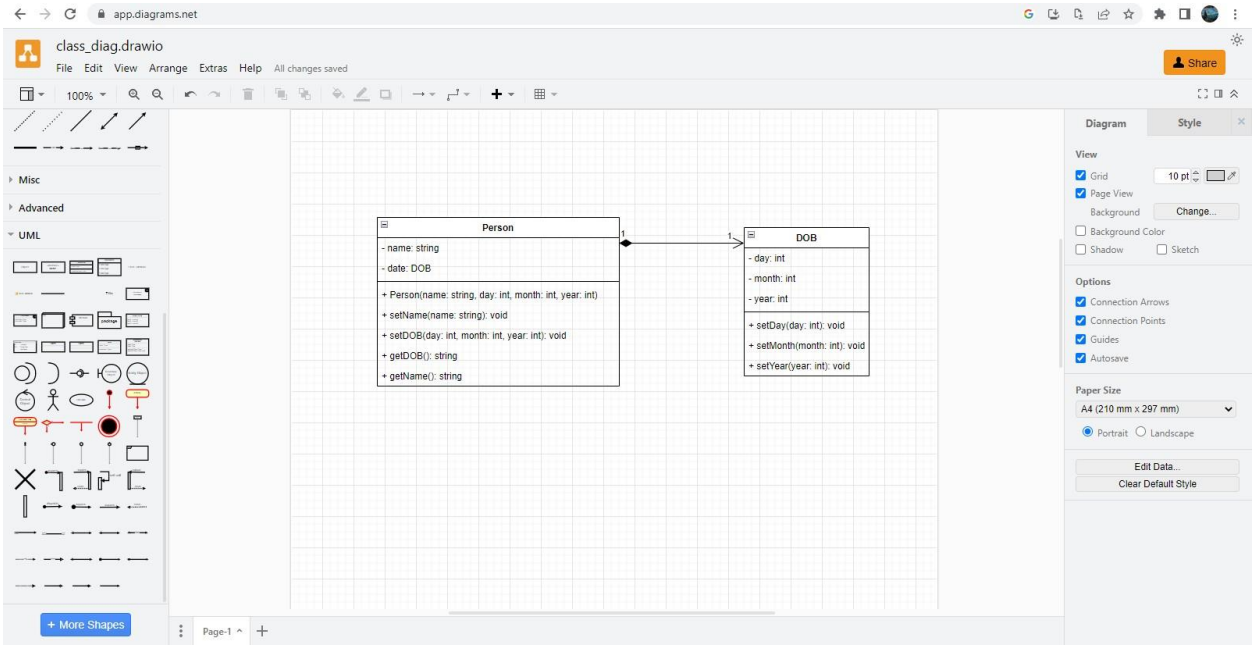
Output:

khan

28/2/1999

Class Diagram in Draw.io

You can create class diagram and other diagrams as well in some tools like draw.io. Open it in your browser, create a new class diagram using the shapes available in the left pane.



To save the diagram, go to File, and then click export (as png, jpeg, or whatever you want), select the option where you want to save.