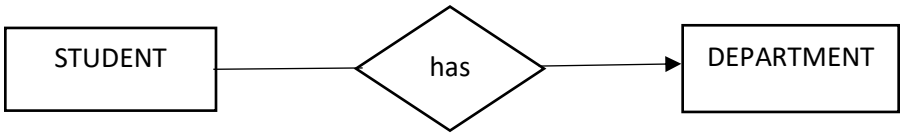
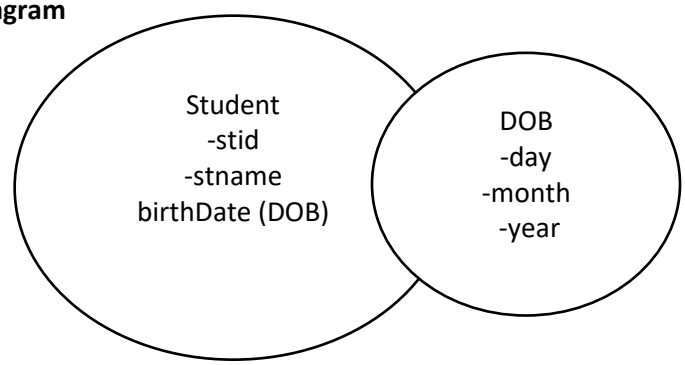
ITEC243 – Lecture Session – 22-Apr-2020

Composition (has-a relationship)



Object Diagram



Composition is also known as has-a relationship between the classes. When a class has an object of another class as a data member, we say that there is composition relationship between them.

Each class should have at least 2 public methods. One of them has to be a constructor (default), and a method that prints the private data members in an appropriate format (or you may create a getter( ) method for each private data members)

Let's see an example

```
//dob.h
class dob{
private:
        int day, year;
        string month;
public:
        dob()
        {
        day = 1;
        month = "January";
        year = 1900;
        }
        dob(int day, string month, int year)
        {
        this->day = day;
        this->month = month;
```

```
        this->year = year;
        }
        int getday() const
        {
        return this->day;
        }
        string getMonth() const
        {
        return this->month;
        }
        int getYear() const
        {
        return this->year;
        }
        void printDOB()
        {
        cout << getday() << "-" << getMonth() << "-" << getYear() << endl;
        }
};


//student.h
class student{
private:
        int stdid;
        string name;
        dob birthdate; //object of "dob" class. here we form has-a relationship between the classes
public:
        student()
        {
        stdid = 101;
        name = "Burak";
        }
        student(int stdid, string name, int day, string month, int year) :birthdate(day, month, year)
        {
        this->stdid = stdid;
        this->name = name;
        }
        void printStudent()
        {
        cout << this->stdid << " " << this->name << " was born on ";
        birthdate.printDOB();
        }
};
```

```
//studentDOB.cpp
#include<iostream>
#include<string>
using namespace std;
#include"DOB.h"
#include"student.h"
void main()
{
	student stobj;
	stobj.printStudent();

	student stobj2(102, "Ayse", 10, "February", 1980);
	stobj2.printStudent();

	system("pause");
}
```

Now let's see how we can pass the object of inner class (DBO) as a parameter inside the outer (student) class.

```
//dob1.h (No changes in this class)
class dob{
private:
	int day, year;
	string month;
public:
	dob()
	{
		day = 1;
		month = "January";
		year = 1900;
	}
	dob(int day, string month, int year)
	{
		this->day = day;
		this->month = month;
		this->year = year;
	}
	int getday() const
	{
		return this->day;
	}
	string getMonth() const
	{
		return this->month;
	}
	int getYear() const
```

```
	{
		return this->year;
	}
	void printDOB()
	{
		cout << getday() << "-" << getMonth() << "-" << getYear() << endl;
	}
};

//student2.h
class student{
private:
	int stdid;
	string name;
	dob birthdate; //object of "dob" class. here we form has-a relationship between the classes
public:
	student()
	{
		stdid = 101;
		name = "Burak";
	}
	student(int stdid, string name, dob bdate):birthdate(bdate)
	{
		this->stdid = stdid;
		this->name = name;
	}
	void printStudent()
	{
		cout << this->stdid << " " << this->name << " was born on ";
		birthdate.printDOB();
	}
};

//studentDOB2.cpp
#include<iostream>
#include<string>
using namespace std;
#include"DOB2.h"
#include"student2.h"
void main()
{
	dob birthd(23, "January", 1879);
	student stobj(1000, "Sebnem", birthd);
	stobj.printStudent();

	system("pause");
}
```