National University of Computer and Emerging Sciences

Lab Manual 09
Object Oriented Programming – CL1004

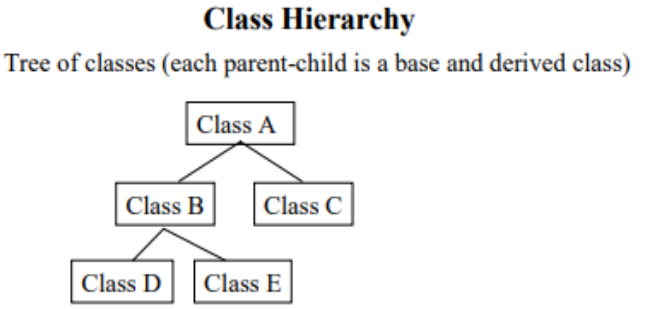
Course Instructor	Dr. Saira Karim
Lab Instructor(s)	Ms. Amna Zulfiqar Mr. Muhammad Adeel
Section	BCS-2B
Semester	Spring 2023
Date	18-04-2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Important Note:

- You may find the syntax to accomplish these exercises from lecture demo.
- Add Necessary Comments in you code to justify your logic.
- Comment exercise number or statement at the start of your code
- Save each exercise in .cpp file with your roll no, ex and lab number e.g.
- 22LXXXX_EX01_Lab01.cpp

- Place all of your exercises in a folder a Zip it (Do not create .rar file) with roll no and lab no. e.g. 22LXXX_Lab01.zip
- Make sure that the interface of your program is user friendly i.e. properly display information.



Lab Manual-09 Inheritance

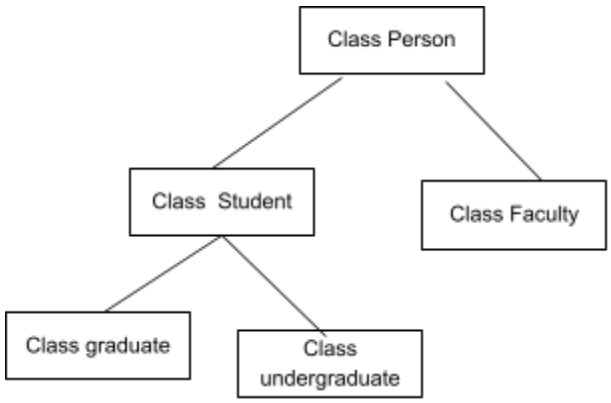
Exercise 1:

The Hierarchy between a parent and child classes looks like the following, where Class A is the parent of Class B and Class C. While Class B is the parent of Class D and Class E

Consider the following hierarchy as it exists in a university as draw its diagram as above:

- There are two types of **persons** in the university i.e. **Student and Faculty**
- Every **Person** has some basic information that is common to all persons i.e. the **first_name** and **last_name** stored as **private** attributes and **age** which is a **protected** attribute.
- A **student** can in turn be either an **Undergraduate** or a **Graduate** student, every student has a **cgpa**.
- An **undergraduate** student has a **fyp_name** as his private attribute.
- A **graduate** student has a **thesis topic** as his private attribute.
- A **faculty** member has private attributes about the number of courses he is currently teaching, i.e. his **course_count** and a three digit telephone extension number i.e. **txt_ext**.

Use the following template and copy paste the box for each class to complete the diagram here :



Exercise 2:

Implement the entire hierarchy that you created in Exercise 1. Every class should be defined in a separate header file named according to the class name.

1. Define all the classes along with their attributes and their inheritance.
2. Add getters and setters for each attribute
3. Add appropriate constructors and destructors to all the classes. For example the constructor for the Person class should take three inputs (for first_name, last_name and age). The student constructor should take four inputs, three for its parent class (i.e. Person) and one float value to be assigned to the cgpa attribute.

This is accomplished in the following manner:

```
Person (char* fname, char* lname, int age)
{
        ...
        cout << "Person() called";
}

Student (char* fname,char* lname,int age,float cgpa): Person(fname,lname,age)
{
        ...
        cout << "Student() called";
}
```

This syntax can be generalized to any parent and child constructor accordingly. Following this syntax, define and implement constructors and destructors for all the classes. Also, Notice that you have to add a print statement in every constructor which announces that the constructor has been called.

Also add a print statement to every destructor which announces that the destructor has been called. For example, the destructor for Person should look like:

```
~Person()
{
        cout << "~Person() called";
}
```

Exercise 3:

Create a C++ source file called 22Lxxxx_Inheritance.cpp. This file contains the main() function. In this main function create an undergraduate student “Muhammad” with cgpa 3.91 who is 22 years of age and a faculty member “Hammad” who is 45 years of age and who is teaching 2 courses this semester and his extension number is 420. Build and execute the code. Copy the Output and Paste it in the section below:

```
Output:
Person (CONSTRUCTOR) called
Student(CONSTRUCTOR) called
Undergraduate(CONSTRUCTOR) called
Person (CONSTRUCTOR) called
Faculty(CONSTRUCTOR) called
Faculty(DESTRUCTOR) called
Person(DESTRUCTOR) called
Undergraduate(DESTRUCTOR) called
Student(DESTRUCTOR) called
Person(DESTRUCTOR) called
```

Exercise 4:

You should notice that the **age** attribute in a Person is **protected**, while the **first_name and last_name** attributes are **private**. What could be the reason for this? Write the reason below:

```
Reason:
So that we could access it from its child classes and change it directly from child class.
```

Now Add a member function **void printInformation()** in the **Person** class. This method should print the name and age of the person.

```
Sample output:
Muhammad is 22 years old
```

Add a member function void **printStudent()** in the **Student** class. This method should print the **name, cgpa and age of the student**. Sample output: “Muhammad is 22 years old, his cgpa is 3.91”

First Use the following Implementation:

```
void Student::printStudent()
{
        cout << first_name << " " << last_name
            << "is " << age <<" years old, his cgpa is " << cgpa;
}
```

Now call the above printStudent() function for the student created in main(). Build the code, you will get an error. Paste the error in the following box.

Error:
member "person::fname" (declared at line 6) is inaccessible
member "person::lname" (declared at line 7) is inaccessible
'person::fname': cannot access private member declared in class 'person'
'person::lname': cannot access private member declared in class 'person'

Why did you get this error?

Reason:
Because first_name and last_name are private members of CLASS Person and hence can't be called directly in CLASS Student.

Exercise 5:

Now change the implementation of the printStudent() function in order to remove the error, but you must still print the required output. Can you use a member function of the base class inside this function? Try that!

Also add a member function void printFaculty() in the Faculty class. This function should print the name, age, number of courses and extension number of the faculty member.

Sample output:

Faculty Member name: Hammad, Age: 45, Number of courses: 2, Ext. 420

Call the printStudent() and printFaculty() in main() in the 22Lxxxx_Inheritance.cpp file. Build and execute the program and Copy the Output and Paste it in the section below.

Output:
muhammad is 22 year old his gpa is 3.91
hammad is 45 year old Course_count 2 Ext: 420

Exercise 6:

Now add two new member functions to the Graduate and Undergraduate Class. These are **void printGraduate() and void printUndergraduate()** respectively.

Their outputs should look like as follows:

Sample output for void printGraduate()
“Ali Hassan is a graduate student, his cgpa is 3.51 and his thesis topic is Distributed Algorithms”

Sample output for void printUndergraduate()
“John is an undergraduate student, his cgpa is 3.01 and his final year project is titled The Even Locator”

Change the main function in 22Lxxxx_Inheritance.cpp to the following:

```
void main(){

        Graduate g                  ("Ali","Hassan",22,3.51,"Distributed
        Algorithms"); Undergraduate        u ("John","Cena",25,3.01,"
        The Even Locator");


        u.printUndergraduate();
        g.printGraduate();

        u.printStudent();
}
```

Observe the output of the above code and paste in below:

Output:
John Cena is 25 year old his gpa is 3.01 and his final year project is titled: The Even Locator
Ali Hassan is 22 year old his gpa is 3.51 and his thesis topic is: Distributed Algorithms
John Cena is 25 year old his gpa is 3.01

Exercise 7:

Now change the *inheritance type* of the *Undergraduate* (which is inheriting from Student) to ***protected*** (previously it was **public**)

```
class Undergraduate : protected Student
{
        ...
};
```

Build the code again. Do you get any errors? Paste the error message in the following box.

Error:
function "student::printStudent" (declared at line 51) is inaccessible
'student::printStudent' not accessible because 'undergraduate' uses 'protected' to inherit from 'student'
'student::printStudent': cannot access protected member declared in class 'undergraduate'

Why did you get this error?

Because now the PUBLIC members of CLASS student are PROTECTED members in CLASS undergraduate and PROTECTED members can only be accessed in child class and friend function or friend class only.

Now **change the inheritance type to private** and build the code again. Do you get any errors? Paste the error message in the following box.

Error:
function "student::printStudent" (declared at line 51) is inaccessible
'student::printStudent' not accessible because 'undergraduate' uses 'private' to inherit from 'student'
'student::printStudent': cannot access private member declared in class 'undergraduate'

Why did you get this error?

Because now the PUBLIC members of CLASS student are PRIVATE members in CLASS undergraduate and PRIVATE members can’t be accessed in friend function or friend class.

Paste the code of your header files here:

```
#include<iostream>

using namespace std;

	class person

	{

	private:

		string  fname;

		string  lname;

	protected:

		int age;

	public:

		person()

		{

			fname = "0";

			lname = "0";

			age = 0;

		}

		person(string f,string l,int a)

		{

			fname =f;

			lname = l;

			age = a;

			cout << "Person (CONSTRUCTOR) called" << endl;

		}

		~person()

		{

			cout << "Person(DESTRUCTOR) called" << endl;
```

```
        }
        void printInformation()
        {
                cout << fname <<" "<<lname << " is " << age << " year old";
        }
};
class student:public person
{
private:
        float cgpa;
public:
        student()
        {
                cgpa = 0;
        }
        ~student()
        {
                cout << "Student(DESTRUCTOR) called" << endl;
        }
        student(string f, string  l, int a, float c) : person(f, l, a)
        {
                cout << "Student(CONSTRUCTOR) called";
                cgpa = c;
        }
        void printStudent()
        {
```

```
                printInformation(); cout << " his gpa is " << cgpa;
        }
};
class graduate:public student
{
private:
        string  thesistopic;
public:
        graduate()
        {
                thesistopic = "0";
        }
        graduate(string  f, string  l, int a, float c,string t):student(f,l,a,c)
        {
                cout << "Graduate(CONSTRUCTOR) called" << endl;
                thesistopic = t;
        }
        ~graduate()
        {
                cout << "Graduate(DESTRUCTOR) called" << endl;
        }
        void printGraduate()
        {
                printStudent(); cout << " and his thesis topic is: " <<
thesistopic<<endl;
        }
};
```

```
class undergraduate:public student
{
private:
	string  fyp_name;
public:
	undergraduate()
	{
		fyp_name = "0";
	}
	~undergraduate()
	{
		cout << "Undergraduate(DESTRUCTOR) called" << endl;
	}
	undergraduate(string f, string  l, int a, float c,string fyp) :student(f, l, a, c)
	{
		cout << "Undergraduate(CONSTRUCTOR) called" << endl;
		fyp_name = fyp;
	}
	void printUndergraduate()
	{
		printStudent(); cout << " and his final year project is titled: " <<
fyp_name<<endl;
	}
};
class faculty:public person
{
private:
```

```
        int course_count;
        int txt_ext;
    public:
        faculty()
        {
            course_count = 0;
            txt_ext = 0;
        }
        faculty(string f,string l,int a,int c,int t):person(f,l,a)
        {
            cout << "Faculty(CONSTRUCTOR) called" << endl;
            course_count = c;
            txt_ext = t;
        }
        ~faculty()
        {
            cout << "Faculty(DESTRUCTOR) called" << endl;
        }
        void printfaculty()
        {
            printInformation(); cout << " Course_count " <<course_count<<"
Ext: "<<txt_ext<<endl;
        }
    };
    int main()
    {
            graduate g("Ali", "Hassan", 22, 3.51, "Distributed Algorithms");
```

```
    undergraduate u("John", "Cena", 25, 3.01, " The Even Locator");
    u.printUndergraduate();
    g.printGraduate();
    u.printStudent();
}
```

Paste the Code of your Source File here: