# National University of Computer and Emerging Sciences



**Lab Manual 09**

# Object Oriented Programming

|  |  |
| --- | --- |
| Course Instructor | Mr. Waqas Manzoor |
| Lab Instructor (s) | Hira Ilyas  Sidra Zafar |
| Section | BSE –A |
| Semester | Spring 2022 |

Department of Computer Science FAST-NU, Lahore, Pakistan

***Inheritance***

**Exercise 1:**

Consider the following hierarchy as it exists in a university:

* 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.

Draw Class Diagram (on paper) to understand the hierarchy of classes that you see in the description above. Include all the attributes in your diagram.

**Exercise 2:**

Implement the entire hierarchy of the Class diagram you created in Exercise 1 i.e. define all the classes along with their attributes and their inheritance. Every class should be defined in a separate header file named according to the class name.

**Exercise 3:**

Add appropriate constructors and destructors to all the classes created in Exercise 2. 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”;

}

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 4 (Optional):**

Add getters and setters for all attributes in all the classes that you have defined.

**Exercise 5:**

Create a C++ source file called **Lab\_Inheritance.cpp**. This file contains the **main()** function.

In this main function create an undergraduate student “Ted Thompson” with cgpa 3.91 who is 22 years of age and a faculty member “Richard Karp” 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 inside /\* comments \*/ in your **Lab\_Inheritance.cpp** fil; like this:

/\*

Output for Exercise 5:

~Person() called

...

...

\*/

**Exercise 6:**

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

**Sample output:** “Ted Thompson 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:** “Ted Thompson is 22 years old, his cgpa is 3.91”

Try to use the following implementation for this function.

void Student::printStudent()

{

cout << first\_name << ” ” << last\_name

<< ”is ” << age <<” years old, his cgpa is ” << cgpa;

}

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

|  |
| --- |
|  |

Why did you get this error?

|  |
| --- |
|  |

**Exercise 7:**

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: Richard Karp, Age: 45, Number of courses: 2, Ext. 420”

Use the following **main()** in the **Lab\_Inheritance.cpp** file. Build and execute the program and paste the output inside comments in the file **Lab\_Inheritance.cpp**.

void main()

{

Student s("Ted","Thompson",22,3.91);

Faculty f("Richard","Karp",45,2,420);

//here the number of courses is 2

//and the extension number is 420

s.printStudent();

f.printFaculty();

}

**Exercise 8:**

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

Their outputs should look like as follows:

**Sample output for void printGraduate()**

“Ted Thompson is a graduate student, his cgpa is 3.91 and his thesis topic is Distributed Algorithms”

**Sample output for void printUndergraduate()**

“Ted Thompson is an undergraduate student, his cgpa is 3.91 and his final year project is titled The Even Locator”

Use the following main function in **Lab\_Inheritance.cpp** to the following:

void main()

{

Undergraduate u ("Ted","Thompson",22,3.91,"The Event Locator");

Graduate g ("Arnold","Gates",25,3.01,"Distributed Algorithms");

u.printUndergraduate();

g.printGraduate();

u.printStudent();

}

Build and execute the program.

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.

|  |
| --- |
|  |

Why did you get this error?

|  |
| --- |
|  |

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.

|  |
| --- |
|  |

Why did you get this error?

|  |
| --- |
|  |