

National University of Computer and Emerging Sciences



Lab Manual Object Oriented Programming

Course Instructor	Dr. Saira Karim
Lab Instructor (s)	Ms. Mamoonah Akbar Ms. Sonia Anum
Section	BSCS 2A
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- Base class derived class concept
- Protected access specifier
- Function overriding
- Types of inheritance

TASK 1:

Exercise 1:

Define and implement a class Shape having data member length, width and having getter and setter functions.

- A default constructor of Shape should print “Shape() called” on the screen.
- A destructor which prints “~Shape () called” on the screen.

Define and implement a class Square in files Square.h and Square.cpp, respectively. This class should be inherited from the Shape class.

- Having data member area and volume
- A default constructor of Square should print “Square() called” on the screen.
- A display function should be there in Square Class which prints its coordinates.
- An area and volume function which print area and volume of square.
- A destructor which prints “~Square () called” on the screen.

Exercise 2:

Define and implement another child class of Shape called “Rectangle”. Complete the class in Rectangle.h and Rectangle.cpp, respectively.

Rectangle class should also have the following features:

- Having data member area and volume
- A default constructor of Rectangle should print “Rectangle () called” on the screen.
- A display function should be there in Rectangle Class which prints its coordinates.
- An area and volume function which print area and volume of rectangle.
- A destructor which prints “~Rectangle () called” on the screen.

TASK 2:

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* and **supervisor name** as his private attributes.
 - A **graduate** student has a *thesis topic* and **supervisor name** as his private attributes.
 - A student can in turn be either an Undergraduate or a Graduate student, every student has a **cgpa** and **rollNumber** as his private attributes.
 - 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.
-
- An **undergraduate** class should be inherited from the person class.
 - A **graduate** class should be inherited from the person and undergraduate class.
 - A **faculty** class should be inherited from the person class.
 - A faculty class should have following functions.
 - Get_age
 - Get_course_count
 - Get_extension
 - Set_extension
 - Set_course_count
 - An undergraduate class should have following functions.
 - ✓ Get_age
 - ✓ Set_cgpa
 - ✓ Get_cgpa
 - ✓ Get_fyp_name
 - ✓ Set_fyp_name
 - ✓ Set_roll_number
 - ✓ Get_roll_number
 - ✓ Set_supervisor_name
 - ✓ Get_supervisor_name
 - A graduate class should have following functions.
 - ✓ Get_age
 - ✓ Set_cgpa
 - ✓ Get_cgpa
 - ✓ Get_thesis_name
 - ✓ Set_thesis_name
 - ✓ Set_roll_number
 - ✓ Get_roll_number
 - ✓ Set_supervisor_name
 - ✓ Get_supervisor_name
 - A person class should have following functions.
 - ✓ Set_age
 - ✓ Set_first_name
 - ✓ Get_first_name
 - ✓ Set_last_name
 - ✓ Get_last_name

Implement these classes 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.