

National University of Computer and Emerging Sciences



Lab Manual # 09 Object oriented programming

Course Instructor	Ms. Arooj Khalil
Lab Instructor(s)	Mamoona Akbar Saleha Batool
Section(s)	BSE-2B1 BSE-2B2
Semester	Spring 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Perform function overriding.
- ✓ Implement function overloading.
- ✓ Call procedure of the parameterized constructor of the base class through derived class.
- ✓ Call procedure of the copy-constructor of the base class through derived class.
- ✓

TASK 1

Exercise 1:

Consider the following hierarchy as it exists in a university:

- There is only one type of person in the university i.e. Student.
- Every Person has some basic information that is common to all persons i.e. the name (a character array), ID and age stored as attributes.
- A student can in turn be either an Undergraduate or a Graduate student, every student has a cgpa and rollNumber.
- An undergraduate student has a fyp_name as his private attribute.
- A graduate student has a thesis topic and supervisor name as his private attribute.

Exercise 2:

1. Add appropriate **constructors** and **destructors** to all the classes. For example, the constructor for the Person class should take three inputs (*ID*, first_name, last_name and age).

- Print “In constructor of class x” and “In Destructor of class x” in constructors and destructors.

2a. Add a member function, **void print()** in the Person class that prints name of person.

2b. Add a member function, **void print()** 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”

2c. Add a member function **void print()** in the Graduate class. This method should print the name, cgpa and age of the student, his thesis topic and supervisor’s name.

Sample output for void print(): “Ted Thompson is a graduate student, his cgpa is 3.91 and his thesis topic is Distributed Algorithms under supervision of Dr. Kashif Zafar”

2d. Add a member function **void print()** in the UnderGraduate class. This method should print the name, cgpa and age of the student and his fyp_name.

Sample output for void print(): “Ted Thompson is an undergraduate student, his cgpa is 3.91 and his final year project is titled The Even Locator”

3a. Add a member function, **void input()** in the Student class. This method should take input in ID, name, cgpa rollNumber and age of the student.

3b. Add a member function **void input()** in the Graduate class. This method should take input in thesis topic, supervisor's name.

3c. Add a member function **void input()** in the UnderGraduate class. This method should take input in fyp_name.

Exercise 3:

Implement this main function:

1. Create two objects UnderGraduate and Graduate class individually.
2. Call the input function to take input in the data from user.
3. Create two pointers of class Student which will hold two objects of UnderGraduate and Graduate class. Call Input and Print functions.
4. Create a pointer of Person class which will hold an object from UnderGraduate/Graduate class (upon your wish). Call Print Function. Notice the output. How would you justify this behavior?

```
//example main function

void main()
{
    //format (ID, fName, lName, Age, RollNumber, cgpa, fyp/thesis)
    Student * s = new Undergraduate(1,"Ted", T"hompson",22, "14L-4171", 3.91,"The Event
Locator");
    Student * s2 = new Graduate(2,"Arnold", "Gates",25, "17L-6171" 3.01,"Distributed
Algorithms");

    s->print();
    s->input();
    s2->print();
    s2->print();
    s->print();
    s2->print();
}
```

Exercise 4:

Implement another function void ChangeInfo() for Graduate Class.

1. Void ChangeInfo(char * ThesisTopic) will set the Thesis Topic to the value passed in parameters.

2. Void `ChangeInfo(char *ThesisTopic, char*Supervisor)` will set both theis topic and Supervisor's name for a graduate student.

Call both functions from main. How do you know which instruction calls which function? Name this phenomenon in comments of your code.

TASK 2

Exercise 1:

Consider a library system that manages different types of library items, including Books, Journals, and Magazines. Implement a class hierarchy using inheritance to represent the different types of library items, and override appropriate member functions to demonstrate the concepts of function overriding and sub-typing details. Additionally, implement a class for `LibraryPatron` who can borrow and return library items.

- You have to create a base class `LibraryItem` and three derived classes `Book`, `Journal`, and `Magazine`.
- The `LibraryItem` class should have `title`, `author`, and `year` as protected member variables, and `setTitle()`, `setAuthor()`, `setYear()`, and `displayDetails()` as member functions.
- The derived classes `Book`, `Journal`, and `Magazine` should inherit from the `LibraryItem` class using different types of inheritance (public, protected, and private respectively).

Exercise 2:

- You have to override the `displayDetails()` function in each of the derived classes to display specific details of the library items.
- Create a `LibraryPatron` class that represents a library patron who can borrow and return library items. The `LibraryPatron` class has a `name` as a private member variable, and `borrowItem()`, `returnItem()`, and `displayBorrowedItems()` as member functions.

Exercise 3:

- In the `main()` function, create objects of the derived classes `Book`, `Journal`, and `Magazine`, set specific properties using member functions, and display the details of the library items using the `displayDetails()` function.
- Also create a `LibraryPatron` object, borrow library items using the `borrowItem()` function, and display the borrowed items using the `displayBorrowedItems()` function.
- Finally return a library item using the `returnItem()` function and display the borrowed items again to demonstrate the functionality of borrowing and returning library items by a library patron.

```
//example main function
```

```
int main()
{
    // Create LibraryItems
    Book book1("Book 1", "Author 1", 2000, "Fiction");
    Journal journal1("Journal 1", "Author 2", 2010, "Science");
    Magazine magazine1("Magazine 1", "Author 3", 2022, "Publisher 1");

    // Display details of LibraryItems
    cout << "--- Book ---" << endl;
    book1.displayDetails();
    cout << endl;

    cout << "--- Journal ---" << endl;
    journal1.displayDetails();
    cout << endl;

    cout << "--- Magazine ---" << endl;
    magazine1.displayDetails();
    cout << endl;

    // Create LibraryPatron
    LibraryPatron patron1("Patron 1");

    // Borrow LibraryItems
    patron1.borrowItem(&book1);
    patron1.borrowItem(&journal1);
    patron1.borrowItem(&magazine1);

    // Display borrowed items by LibraryPatron
    patron1.displayBorrowedItems();
    cout << endl;

    // Return LibraryItem
    patron1.returnItem(&journal1);

    // Display borrowed items by LibraryPatron after returning an item
    patron1.displayBorrowedItems();

    return 0;
}
```