Lahore University of Management and Sciences

Lab-8 Manual

for

Introduction to Programming

(CS200)

Dr. Mian Muhammad Awais

LAB GUIDELINES

- Make sure you submit the lab before 11:50 AM. Any late submission will not be graded afterwards. In case of internet connectivity or electricity issues make sure to email your assigned TA before 2:00 PM. **You should email your assigned TA ONLY**.
- For every lab, there will be a folder created on LMS. You must submit your work in the respective folder during the lab time, you and only you are responsible for your submissions.
- You will be allowed to discuss the questions in the first half of the lab session for a few questions. After that, there will be a portion of lab where you cannot converse and must work for yourself. No discussion is allowed in later time period.
- You should do your work with utmost clarity and precision. Do not waste your time trying to do something you do not understand. Ask Lab instructors for help, that is what they are there for.

- Any legitimate cheating case can and will be reported to Disciplinary Committee without any leniency. Plagiarism Software make our task easier.
- Please follow the lab etiquettes and follow code of conduct in the session.
- Do not start Personal chat during your zoom meeting and raise your hands before asking questions.
- **Make separate .cpp files for each question. Naming convention for .cpp files is: YourRollNumber_TaskX.cpp. Before submission, copy all the .cpp files in a folder named LabX_AssignedTAName_YourRollNumber.zip. Submit the .zip file only! (no .rar file submissions). X should be replaced by appropriate number.** Failure to follow the naming convention may lead to deduction of marks.

OBJECTIVES

- Inheritance

LAB EXERCISES

Question # 1 [Marks: 40] Est. Time: 30 mins

In this task, you will be implementing multiple inheritance. You will have 4 classes:

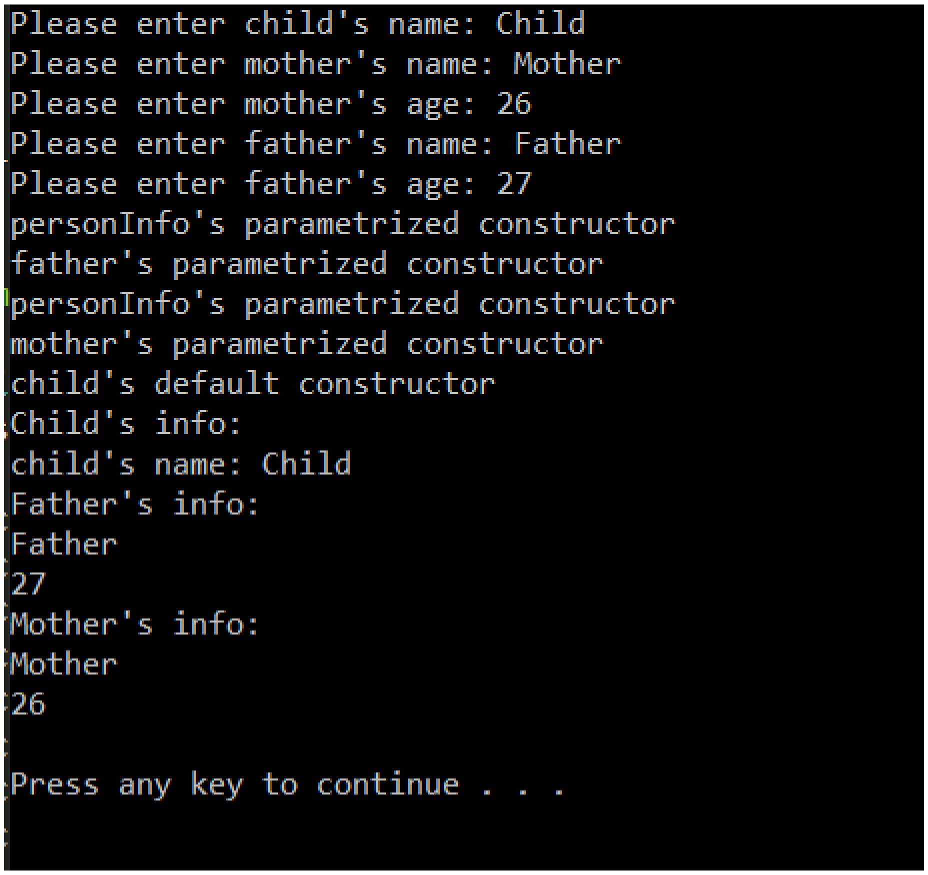
- **class personInfo**
  - 2 private members: **string name**, **int age**
  - **Constructor (default and parametrized both)**, **setters and getters** for all data members and a **printPersonInfo** function that prints the value of data members with appropriate prompts (look at sample output for the prompts)
- **class father**
  - Inherits publicly from personInfo class
  - Has no data members of its own
  - **Constructor (default and parametrized both)** and a **printFather** function that makes a call to parent's print function. You are required to add appropriate prompts in this print function as well (look at sample output for the prompts)
- **class mother**
  - Inherits publicly from personInfo class
  - Has no data members of its own
  - **Constructor (default and parametrized both)** and a **printMother** function that makes a call to parent's print function. You are required to add appropriate prompts in this print function as well (look at sample output for the prompts)

- **class child**
  - Inherits publicly from both father and mother classes
  - 1 private member: **string Name**
  - Constructor (default and parametrized both) and a **printChild** function that not only prints this class' members but also makes call to parents' print functions. You are required to add appropriate prompts in this print function as well (look at sample output for the prompts)

Please copy the following code and paste it in the int main (This code MUST NOT be changed):

```
int main()

{

        int age1, age2 = 0;

        string name1, name2, name3 = "";

        cout  << "Please enter child's name: ";

        cin >> name1;

        cout  << "Please enter mother's name: ";

        cin >> name2;

        cout  << "Please enter mother's age: ";

        cin >> age1;

        cout  << "Please enter father's name: ";

        cin >> name3;

        cout  << "Please enter father's age: ";

        cin >> age2;

        child obj(name1, name3,  age2, name2, age1);

        obj.printChild();

        return 0;

}
```

Sample Output



```
Please enter child's name: Child
Please enter mother's name: Mother
Please enter mother's age: 26
Please enter father's name: Father
Please enter father's age: 27
personInfo's parametrized constructor
father's parametrized constructor
personInfo's parametrized constructor
mother's parametrized constructor
child's default constructor
Child's info:
child's name: Child
Father's info:
Father
27
Mother's info:
Mother
26

Press any key to continue . . .
```

Marks' distribution:

Perfect skeleton for all 4 classes: 20 (5 per class)

Correct Inheritance relation setup between all 4 classes: 10

Corrects function calls made using these inheritance relations: 10

Question # 2 [Marks: 60] Est. Time: 60 mins

The Library Management System at LUMS is inefficient, probably because of the redundancy in its code.
The management system consists of three C++ classes, their details are as follows:

- **class Manuscripts**
  - Methods
    - Constructors (default + parametrized)

        - Getters/Setters
        - print()
    - Member Variables
        - ID (int)
        - isIssued (bool)
        - manuscriptName (string)
- **class Newspaper**
    - Methods:
        - Constructors (default + parametrized)
        - Getters/Setters
        - print()
    - Member Variables:
        - ID (int)
        - IsIssued (bool)
        - newsPaperTitle (string)
- **class Books**
    - Methods:
        - Constructors (default + parametrized)
        - Getters/Setters
        - print()
    - Member Variables:
        - ID (int)
        - isIssued (bool)
        - bookName (string)
        - bookAuthorName (string)

Your task is to alter this implementation and remove the redundancy using concepts from your lectures on inheritance.

To test your implementation, create objects of all child classes in the main function, use setters to alter the variables and getters to access them in main ().
Use your print function to print the values and show that the getters and setters are working properly.

NOTE: There is no one correct output to this question.

Marks' Distribution:
Correct structure of all classes: 20
Making a correct and efficient implementation for the library management system using inheritance: 40

Best of Luck!