Lab 4: Classes, Objects, and Object Interaction in C++

Lab Instructions

- Write all programs in C++
- Follow proper class design principles
- Clearly separate private data members and public member functions
- Use meaningful class names, object names, and function names
- Add comments to explain logic and observations
- Each exercise must be written as a separate program

Exercise 1: Create a class named `Student`.

- Declare the following data members as **private**:
  - `rollNo` (int)
  - `marks` (int)
- Declare the following member functions as **public**:
  - `setData()` to assign values to the data members
  - `display()` to print student details

In `main()`:

- Create an object named `s1`
- Store and display the student details using public member functions

Exercise 2: Create a class named `Number`.

- Declare one private data member:
  - `value` (int)
- Declare a **default constructor** that initializes `value` to 10
- Declare a public member function `show()` to display the value

In `main()`:

- Create an object named `n1`

- Display the initialized value

Exercise 3: Create a class named `Rectangle`.

- Declare the following data members as **private**:
  - `length` (int)
  - `width` (int)
- Declare the following member functions as **public**:
  - `setDimensions(int l, int w)`
  - `calculateArea()`
  - `displayArea()`

In `main()`:

- Create an object named `rect1`
- Set dimensions and display the area

Exercise 4: Create a class named `BankAccount`.

- Declare the following data member as **private**:
  - `balance` (double)
- Declare the following member functions as **public**:
  - `setBalance(double amount)`
  - `getBalance()`

In `main()`:

- Create an object named `acc1`
- Set and retrieve the balance using public functions only
- Do not access the private data member directly

Exercise 5: Create a class named `Employee`.

- Declare the following data members as **private**:
  - `empId` (int)
  - `salary` (float)

- Declare the following member functions as **public**:
  - `setData(int id, float sal)`
  - `showData()`

In `main()`:

- Create an array of three objects named `emp[3]`
- Store and display employee details using loops

Exercise 6: Create two classes named `Author` and `Book`.
Class Author

- Private data member:
  - `name` (string)
- Public member functions:
  - `setName(string n)`
  - `getName()`

Class Book

- Private data member:
  - `title` (string)
- Public member function:
  - `display(Author a)` to display book title and author name

In `main()`:

- Create objects `a1` (Author) and `b1` (Book)
- Pass the Author object to the Book object

Exercise 7: Create a class named `Demo`.

- Declare a private data member:
  - `num` (int)
- Declare public member functions:
  - `set(int num)` using the `this` pointer
  - `show()` to display the value

In `main()`:

- Create an object named `d1`
- Call both functions

Exercise 8: Create a class named `Sample`.

- Declare a public data member:
  - `x` (int)

Write a function named `modify()`:

- Accepts a `Sample` object by value
- Modifies the value of `x`

In `main()`:

- Create an object named `obj1`
- Show that changes inside the function do not affect the original object
- Explain why the object address and its copy are different

Exercise 9: Using the same `Sample` class from Exercise 8:

- Write a function named `modifyByAddress()`
- Accept a pointer to a `Sample` object
- Modify the data member using pointer notation

In `main()`:

- Create an object named `obj2`
- Pass its address to the function
- Show that the changes affect the original object
- Access object data using both object and pointer