

Lab 08 Tasks

Task 01

You are designing a library system using C++. You have a class called "Book" with attributes like `title`, `author`, and `publisher`. Now, you want to create a derived class called "FictionBook" which inherits from the "Book" class and has additional attributes like `genre` and `protagonist`.

- a) Define the "Book" class with the following attributes:
 - `title (string)`
 - `author (string)`
 - `publisher (string)`
 - Define the constructor for the "Book" class which initializes the attributes.
 - Define a function called "display" inside the "Book" class which displays the details of the book.
- b) Create a derived class called "FictionBook" which inherits from the "Book" class and has the following additional attributes:
 - `genre (string)`
 - `protagonist (string)`
 - Define the constructor for the "FictionBook" class which initializes the attributes of the base class as well as the derived class.
 - Define a function called "display" inside the "FictionBook" class which displays the details of the book, including the additional attributes.
 - In the `main` function, create an object of the "FictionBook" class and display its details.

Task 02

Suppose you are developing a game that has different types of characters. You have a base class called "Character" which has attributes like `health` and `damage`. Now, you want to create two derived classes - "Enemy" and "Player" - which inherit from the "Character" class. Further, you want to create another class called "Wizard" which inherits from the "Player" class and has additional attributes like `magic power` and `spells`.

- a) Define the "Character" class with the following attributes:
 - `health (integer)`
 - `damage (integer)`
 - Define the constructor for the "Character" class which initializes the attributes.
 - Define a function called "display" inside the "Character" class which displays the details of the character.
- b) Create a derived class called "Enemy" which inherits from the "Character" class.
 - Define the constructor for the "Enemy" class which initializes the attributes of the base class.
 - Define a function called "display" inside the "Enemy" class which displays the details of the enemy.

- c) Create a derived class called "Player" which inherits from the "Character" class.
- Define the constructor for the "Player" class which initializes the attributes of the base class.
 - Define a function called "display" inside the "Player" class which displays the details of the player.
- d) Create another derived class called "Wizard" which inherits from the "Player" class and has the following additional attributes:
- magic power (integer)
 - spells (string)
 - Define the constructor for the "Wizard" class which initializes the attributes of the base classes as well as the derived class.
 - Define a function called "display" inside the "Wizard" class which displays the details of the wizard, including the additional attributes.

In the `main` function, create an object of the "Wizard" class and display its details.

Task 03

Suppose you are developing a game engine that involves a variety of game objects, such as characters, enemies, and items. You want to implement a system to keep track of the position of these objects in 3D space, as well as their health status.

To accomplish this, you decide to create two base classes: `Position` and `Health`. The `Position` class will have three member variables to store the `x`, `y`, and `z` coordinates of an object, and the `Health` class will have a single member variable to store the health of an object.

You also want to create a `Character` class that inherits from both `Position` and `Health`. This class will represent characters in the game, and will have additional member variables and member functions to handle character-specific actions.

Write the code to define the `Position`, `Health`, and `Character` classes, and demonstrate how to create and use an object of the `Character` class.

Task 04

Suppose you are developing a program for a school that needs to keep track of its students and teachers. Both students and teachers have a `name` and an `age`, but they have different properties beyond that. Students have a `student ID` number and a `grade level`, while teachers have a `subject` they teach and a `room number`.

You want to create a set of classes to represent these objects, using hybrid inheritance. Specifically, you want to create a base class called `Person` that has member variables for the `name` and `age`, a derived class called `Student` that inherits from `Person` and adds member variables for the `student ID` and `grade level`, and a derived class called `Teacher` that inherits from `Person` and adds member variables for the `subject` and `room number`.

You also want to create a derived class called `GraduateStudent` that inherits from both `Student` and `Teacher`, representing a student who is also a teacher's assistant.

Write the code to define the `Person`, `Student`, `Teacher`, and `GraduateStudent` classes, and demonstrate how to create and use an object of the `GraduateStudent` class.

Task 05

Suppose you are creating a program to represent different types of vehicles. You want to create a base class called `Vehicle` that has member variables for the `make`, `model`, and `year`, as well as a constructor to initialize these variables. You then want to create a derived class called `Car` that inherits from `Vehicle` and adds member variables for the number of `doors` and the `fuel efficiency`. Finally, you want to create a derived class called `ElectricCar` that inherits from `Car` and adds a member variable for the `battery life`.

Write the code to define the `Vehicle`, `Car`, and `ElectricCar` classes, including constructors to initialize the member variables. In each derived class constructor, make sure to call the appropriate base class constructor using an initialization list. Also, demonstrate how to create and use an object of the `ElectricCar` class.