**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Lab 11 | |
| **Topic** | Inheritance |
| **Objective** | demonstrates the concept of inheritance and its types. |

**Task 1: Inheritance in C++**

**Base Class: Vehicle**

* + Private Member Variables:
    - **model** (string): to store the vehicle model.
    - **year** (int): to store the manufacturing year.
  + Public Member Functions:
    - Default constructor: Initializes the model as "Undefined" and the year as 0.
    - Parameterized constructor: Accepts parameters for the model and year and initializes the **Vehicle**.
    - **display()**: Displays the model and year of the vehicle.

**Derived Class 1: Car**

* + Inherits publicly from **Vehicle**.
  + Additional Private Member Variable:
    - **numDoors** (int): to store the number of doors.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets **numDoors** to 4.
    - Parameterized constructor: Accepts parameters for the model, year, and number of doors and initializes the **Car**.
    - **display()**: Extends the base class **display()** to include the number of doors.

**Derived Class 2: Motorcycle**

* + Inherits privately from **Vehicle**.
  + Additional Private Member Variable:
    - **numWheels** (int): to store the number of wheels.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets **numWheels** to 2.
    - Parameterized constructor: Accepts parameters for the model, year, and number of wheels and initializes the **Motorcycle**.
    - **display()**: Extends the base class **display()** to include the number of wheels.

**Derived Class 3: Truck**

* + Inherits protectedly from **Vehicle**.
  + Additional Private Member Variable:
    - **cargoCapacity** (double): to store the cargo capacity.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets **cargoCapacity** to 0.0.
    - Parameterized constructor: Accepts parameters for the model, year, and cargo capacity and initializes the **Truck**.
    - **display()**: Extends the base class **display()** to include the cargo capacity.

**Task Requirements:**

* Implement the base class **Vehicle** and the derived classes **Car**, **Motorcycle**, and **Truck** with the specified features.
* Demonstrate the use of public, private, and protected inheritance for the derived classes.
* Create objects of each class and display their information using the **display()** method.
* Write a concise explanation of how public, private, and protected inheritance affects the access levels of the base class members in the derived classes.

**Task 2: Shape Hierarchy**

Create a C++ program to model a hierarchy of geometric shapes. Design a base class called **Shape** and two derived classes: **Circle** and **Rectangle**. Use different types of inheritance for the derived classes. Each class should have specific attributes and methods.

* **Base Class: Shape**
  + Private Member Variables:
    - **color** (string): to store the color of the shape.
  + Public Member Functions:
    - Default constructor: Initializes the color as "Undefined".
    - Parameterized constructor: Accepts a parameter for the color and initializes the **Shape**.
    - **void display()**: Displays the color of the shape.
* **Derived Class 1: Circle**
  + Inherits publicly from **Shape**.
  + Additional Private Member Variable:
    - **radius** (double): to store the radius of the circle.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets the radius to 0.0.
    - Parameterized constructor: Accepts parameters for the color and radius and initializes the **Circle**.
    - **void display()**: Extends the base class **display()** to include the radius.
* **Derived Class 2: Rectangle**
  + Inherits privately from **Shape**.
  + Additional Private Member Variables:
    - **length** (double): to store the length of the rectangle.
    - **width** (double): to store the width of the rectangle.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets the length and width to 0.0.
    - Parameterized constructor: Accepts parameters for the color, length, and width, and initializes the **Rectangle**.
    - **void display()**: Extends the base class **display()** to include the length and width.

**Task Requirements:**

* Implement the base class **Shape** and the derived classes **Circle** and **Rectangle** with the specified features.
* Use different types of inheritance for the derived classes.
* Create objects of each class and display their information using the **display()** method.
* Observe how different types of inheritance affect the access to base class members.

**Task 3: Employee Hierarchy**

Create a C++ program to model a hierarchy of employees in a company. Design a base class called **Employee** and three derived classes: **Manager**, **Engineer**, and **Clerk**. Use different types of inheritance for the derived classes. Each class should have specific attributes and methods.

* **Base Class: Employee**
  + Private Member Variables:
    - **name** (string): to store the name of the employee.
  + Public Member Functions:
    - Default constructor: Initializes the name as "Unknown".
    - Parameterized constructor: Accepts a parameter for the name and initializes the **Employee**.
    - **void display()**: Displays the name of the employee.
* **Derived Class 1: Manager**
  + Inherits publicly from **Employee**.
  + Additional Private Member Variable:
    - **department** (string): to store the department of the manager.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets the department as "Undefined".
    - Parameterized constructor: Accepts parameters for the name and department and initializes the **Manager**.
    - **void display()**: Extends the base class **display()** to include the department.
* **Derived Class 2: Engineer**
  + Inherits privately from **Employee**.
  + Additional Private Member Variable:
    - **project** (string): to store the current project of the engineer.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets the project as "No Project".
    - Parameterized constructor: Accepts parameters for the name and project and initializes the **Engineer**.
    - **void display()**: Extends the base class **display()** to include the project.
* **Derived Class 3: Clerk**
  + Inherits protectedly from **Employee**.
  + Additional Private Member Variable:
    - **task** (string): to store the assigned task of the clerk.
  + Additional Public Member Functions:
    - Default constructor: Calls the base class default constructor and sets the task as "No Task".
    - Parameterized constructor: Accepts parameters for the name and task and initializes the **Clerk**.
    - **void display()**: Extends the base class **display()** to include the task.

**Task Requirements:**

* Implement the base class **Employee** and the derived classes **Manager**, **Engineer**, and **Clerk** with the specified features.
* Use different types of inheritance for the derived classes.
* Create objects of each class and display their information using the **display()** method.
* Observe how different types of inheritance affect the access to base class members