**Task: Implement a C++ Program to Calculate and Display Student Results Using Multiple Inheritance**

**Objective**: Create a C++ program that demonstrates the use of multiple inheritance by simulating a system where a student's academic and sports performance is integrated and displayed.

**Instructions:**

1. **Class Definitions**:
2. **Class student**:
3. **Protected Data Member**:
4. int rollno: Stores the student's roll number.
5. **Public Member Functions**:
6. void setrollno(int a): Sets the roll number of the student.
7. void getrollno(): Displays the student's roll number.
8. **Class test (Inherits from student)**:
9. **Protected Data Members**:
10. float part1, part2: Stores the marks obtained in two parts of a test.
11. **Public Member Functions**:
12. void putmarks(float a, float b): Sets the marks for part 1 and part 2 of the test.
13. void getmarks(): Displays the marks obtained in part 1 and part 2.
14. **Class sports**:
15. **Protected Data Member**:
16. float score: Stores the sports score.
17. **Public Member Functions**:
18. void setscore(float s): Sets the sports score.
19. void getscore(): Displays the sports score.
20. **Class result (Inherits from test and sports)**:
21. **Private Data Member**:
22. float total: Stores the total score which is the sum of part 1, part 2, and the sports score.
23. **Public Member Function**:
24. void display():
25. Calculates the total score.
26. Calls getrollno(), getmarks(), and getscore() to display the roll number, marks, and sports score respectively.
27. Displays the total score.
28. **Main Function**:
29. **Create an object** of the result class.
30. **Use the object to**:
31. Set the student's roll number using setrollno(int a).
32. Set the test marks using putmarks(float a, float b).
33. Set the sports score using setscore(float s).
34. Display the complete result using the display() method, which shows the roll number, marks, sports score, and total score.

**Key Concepts to Understand:**

1. **Inheritance**: The program demonstrates both single and multiple inheritance. The test class inherits from the student class, while the result class inherits from both test and sports.
2. **Encapsulation**: The data members are kept protected/private to ensure encapsulation, allowing access only through public member functions.
3. **Method Integration**: The display() function in the result class integrates data from multiple classes to provide a comprehensive output.

**Expected Output:**

When the code is executed, it should display:

1. The student's roll number.
2. The marks obtained in part 1 and part 2 of the test.
3. The sports score.
4. The total score, which is the sum of part 1, part 2, and the sports score.

**Task: Implement a C++ Program to Calculate and Display Student Results Using Inheritance**

**Objective**: Write a C++ program that demonstrates the concept of single inheritance by calculating and displaying a student's total marks based on their individual subject scores.

**Instructions:**

1. **Class Definitions**:
2. **Class student**:
3. **Protected Data Member**:
4. int rollno: Stores the student's roll number.
5. **Public Member Functions**:
6. void setrollno(int a): Assigns the roll number to the student.
7. void getrollno(): Displays the student's roll number.
8. **Class test (Inherits from student)**:
9. **Protected Data Members**:
10. float sub1, sub2: Stores the marks obtained in two subjects.
11. **Public Member Functions**:
12. void putmarks(float a, float b): Assigns the marks to sub1 and sub2.
13. void getmarks(): Displays the marks obtained in the two subjects.
14. **Class result (Inherits from test)**:
15. **Private Data Member**:
16. float total: Stores the total marks.
17. **Public Member Function**:
18. void display():
19. Calculates the total marks by summing up sub1 and sub2.
20. Calls getrollno() to display the student's roll number.
21. Calls getmarks() to display the marks of both subjects.
22. Displays the total marks.
23. **Main Function**:
24. **Create an object** of the result class named student1.
25. **Use the object to**:
26. Set the student's roll number using setrollno(int a).
27. Set the marks for the two subjects using putmarks(float a, float b).
28. Display the complete result using the display() method, which shows the roll number, marks of each subject, and the total marks.

**Key Concepts to Understand:**

1. **Inheritance**: The result class inherits from the test class, which in turn inherits from the student class. This demonstrates the concept of single inheritance where classes are derived from one another.
2. **Encapsulation**: The data members such as rollno, sub1, and sub2 are protected to ensure encapsulation, meaning they can only be accessed through public member functions.
3. **Integration of Methods**: The display() function integrates data from the student and test classes to provide a comprehensive output of the student's results.

**Expected Output:**

When the code is executed, it should display:

1. The student's roll number.
2. The marks obtained in each of the two subjects.
3. The total marks, which is the sum of the two subject marks.

**Task: Implement a C++ Program Using Multiple Inheritance to Perform Basic Arithmetic Operations**

**Objective**: Write a C++ program that demonstrates the concept of multiple inheritance by calculating and displaying the product of two numbers using two different classes to set their values.

**Instructions:**

1. **Class Definitions**:
2. **Class M**:
3. **Protected Data Member**:
4. int m: Stores a value.
5. **Public Member Functions**:
6. void setm(int x): Assigns the value x to the member m.
7. **Class N**:
8. **Protected Data Member**:
9. int n: Stores a value.
10. **Public Member Functions**:
11. void setn(int y): Assigns the value y to the member n.
12. **Class P (Inherits from both M and N)**:
13. **Public Member Function**:
14. void display():
15. Displays the values of m and n.
16. Calculates and displays the product of m and n.
17. **Main Function**:
18. **Create an object** of the class P named p.
19. **Use the object to**:
20. Set the value of m using setm(int x).
21. Set the value of n using setn(int y).
22. Call the display() method to output:
23. The value of m.
24. The value of n.
25. The product of m and n.

**Key Concepts to Understand:**

1. **Multiple Inheritance**: The P class inherits from both M and N, demonstrating how a class can derive from multiple base classes.
2. **Encapsulation**: The data members m and n are protected, ensuring that they can only be accessed through public member functions.
3. **Integration of Methods**: The display() function integrates data from both base classes to provide a comprehensive output.

**Expected Output:**

When the code is executed, it should display:

1. The value of m.
2. The value of n.
3. The product of m and n.

**Task: Implement a C++ Program Using Class Inheritance and Access Specifiers**

**Objective**: Write a C++ program that demonstrates the use of class inheritance and access specifiers by creating a class hierarchy to perform multiplication and display values.

**Instructions:**

1. **Class Definitions**:
2. **Class B**:
3. **Private Data Members**:
4. int a: Stores a private integer value.
5. **Public Data Members**:
6. int b: Stores a public integer value.
7. **Public Member Functions**:
8. void getab(): Prompts the user to input values for a and b.
9. int geta(): Returns the value of the private member a.
10. void showa(): Displays the value of a.
11. **Class D** (Inherits privately from B):
12. **Private Data Members**:
13. int c: Stores the result of the multiplication.
14. **Public Member Functions**:
15. void mul(): Calls getab() from class B to get the values of a and b, and calculates the product of b and a, storing it in c.
16. void display(): Displays the values of a, b, and c.
17. **Main Function**:
18. **Create an object** of the class D named d.
19. **Use the object to**:
20. Call the mul() method to get input values and perform the multiplication.
21. Call the display() method to output:
22. The value of a.
23. The value of b.
24. The value of c.

**Key Concepts to Understand:**

1. **Private Inheritance**: The class D inherits privately from B, meaning that members of class B cannot be accessed directly from objects of class D.
2. **Encapsulation**: The private member a in class B is accessed through the public member function geta(), demonstrating encapsulation principles.
3. **Data Manipulation**: The program collects user input, performs arithmetic operations, and displays results, showcasing how data can be manipulated across classes.

**Expected Output:**

When the code is executed, it should prompt the user to enter values for a and b, and then display:

1. The value of a.
2. The value of b.
3. The product of a and b (stored in c).

**Task: Implement a C++ Program Using Class Inheritance and Method Overriding**

**Objective**: Write a C++ program that demonstrates the use of class inheritance, method overriding, and data manipulation through member functions.

**Instructions:**

1. **Class Definitions**:
2. **Class B**:
3. **Private Data Members**:
4. int a: Stores a private integer value.
5. **Public Data Members**:
6. int b: Stores a public integer value.
7. **Public Member Functions**:
8. void set(): Initializes the values of a and b (set a to 5 and b to 10).
9. void mul(int c): Takes an integer c as a parameter and calculates the product of b and c, storing the result in a.
10. int display(): Returns the value of a.
11. **Class D** (Inherits publicly from B):
12. **Private Data Members**:
13. int d: Stores an integer value.
14. **Public Member Functions**:
15. void sum(int c): Takes an integer c, assigns it to d, and adds the value of b to d.
16. void display(): Outputs the values of b and d.
17. **Main Function**:
18. **Create an object** of the class D named d.
19. **Use the object to**:
20. Call the set() method to initialize the values of a and b.
21. Call the mul(7) method to calculate the product of b and 7, storing it in a.
22. Output the result of display() from class B using d.B::display().
23. Call the sum(10) method to set d to 10 and add the value of b to d.
24. Output the result of display() from class D using d.D::display().

**Key Concepts to Understand:**

1. **Public Inheritance**: The class D inherits publicly from B, allowing it to access public members of B.
2. **Method Overriding**: The display() method in class D overrides the display() method in class B, allowing for different behaviors based on the context of the object.
3. **Data Manipulation**: The program demonstrates how data can be set, manipulated, and displayed across class hierarchies.

**Expected Output:**

When the code is executed, it should display:

1. The value of a after calculating the product (using d.B::display()).
2. The values of b and d after summing them in the sum method (using d.D::display()).