

Overview

This C++ program implements a set of interactive tasks that users can choose to perform. Each task is encapsulated within its own class, which inherits from a base abstract class called `LabTask`. The available tasks include array comparison, displaying a 10x10 ascending array, performing a Caesar cipher encryption and decryption, and managing student grades. Users can select a task from a menu, and the corresponding functionality is executed.

Key Classes and Components

LabTask

`LabTask` is an abstract base class that defines a pure virtual function `execute()`. All derived classes must implement this function to provide specific functionality for each task.

ArrayComparer

This class implements functionality for comparing two arrays and finding their common elements.

- **Constructor:** Initializes two sample arrays.
- **findCommonElements():** Iterates through the first array and checks for common elements in the second array using the `std::find` algorithm.
- **execute():** Calls the `findCommonElements()` method to display common elements.

AscendingArray

This class generates and displays a 10x10 array filled with ascending integers.

- **Constructor:** Initializes the 10x10 array with values from 1 to 100.
- **displayArray():** Outputs the contents of the array in a formatted manner.
- **execute():** Calls the `displayArray()` method to show the ascending array.

CaesarCipher

This class provides methods for encrypting and decrypting text using the Caesar cipher technique.

- **Constructor:** Accepts a shift value (default is 3).
- **encrypt():** Encrypts the input text by shifting each letter by the specified amount.
- **decrypt():** Decrypts the encrypted text by reversing the shift.
- **execute():** Demonstrates encryption and decryption on a sample string ("HelloWorld").

StudentGrades

This class manages student grades, allowing users to input grades for multiple students and calculate statistics.

- **Constructor:** Initializes a 2D vector to store grades for three students.
- **inputGrades():** Prompts users to enter five grades for each student.
- **calculateSum():** Computes the sum of grades for a given student.
- **calculateAverage():** Calculates the average grade based on total grades.

- **calculateMedian()**: Computes the median of grades after sorting them.
- **displayGrades()**: Displays each student's grades along with their sum, average, and median.
- **execute()**: Calls `inputGrades()` and `displayGrades()` to manage student grade data.

Execution Flow

Main Menu:

The `main()` function presents a menu to the user, allowing them to select one of four tasks:

1. Array Comparison
2. 10x10 Ascending Array
3. Caesar Cipher
4. Student Grades

Task Execution:

Based on user input, the corresponding task's `execute()` method is invoked. Each task handles its own input/output, allowing for independent execution of functionalities.

Program Termination:

After executing a selected task, the program terminates gracefully without prompting for further actions, as there is no loop implemented for repeated task selection.

Example User Interaction

Array Comparison

```
Select a task:
1. Array Comparison
2. 10x10 Ascending Array
3. Caesar Cipher
4. Student Grades
>> 1
Common elements: 5 6 7 8 9 10
```

Ascending Array

```
Select a task:
1. Array Comparison
2. 10x10 Ascending Array
3. Caesar Cipher
4. Student Grades
>> 2
1    2    3    4    5    6    7    8    9    10
11   12   13   14   15   16   17   18   19   20
21   22   23   24   25   26   27   28   29   30
31   32   33   34   35   36   37   38   39   40
41   42   43   44 45 46 47 48 49 50
```

```
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

Caesar Cipher

```
Select a task:
1. Array Comparison
2. 10x10 Ascending Array
3. Caesar Cipher
4. Student Grades
>> 3
Original Text: HelloWorld
Encrypted Text: KoorZruog
Decrypted Text: HelloWorld
```

Student Grades

```
Select a task:
1. Array Comparison
2. 10x10 Ascending Array
3. Caesar Cipher
4. Student Grades
>> 4
Enter 5 grades for student 1: 80 90 85 70 75
Enter 5 grades for student 2: 88 92 95 89 91
Enter 5 grades for student 3: 78 82 80 76 74

Student 1 Grades:
Sum: 400
Average: 80
Median: 80

Student 2 Grades:
Sum: 455
Average: 91
Median: 90

Student 3 Grades:
Sum: 390
Average: 78
Median: 78
```

Enhancements & Improvements

Error Handling:

- Implement error handling for invalid inputs (e.g., non-integer values when entering grades).

Dynamic Input:

- Allow users to dynamically specify how many students or how many grades per student they want to enter.

Additional Features:

- Extend functionality in `CaesarCipher` class to allow user-defined text input instead of hardcoded strings.
- Add sorting capabilities in `StudentGrades` to display students based on their average grade.

Conclusion

This program effectively demonstrates object-oriented programming principles such as inheritance and polymorphism in C++. Each task is encapsulated within its respective class, making it modular and easy to extend or modify in future iterations. With enhancements, this program can become even more user-friendly and robust in handling various educational tasks.