

04. Practice: Enhanced Function Features

Practice Content

■ Practice #4

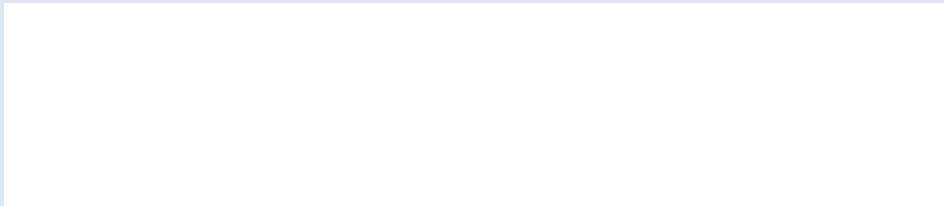
- 4-1. Understanding Function Templates#
- 4-2. Improving Grade Processing Program #2 Using Inline Functions

Practice #4-1: Understanding Function Templates (1)

- Create an add() function that receives an array and its size as parameter
 - add() is a generic function that takes two arguments (array and size).
 - The function should sum all the elements in the array.
 - It should return the computed value.

```
#include <iostream>
using namespace std;
```

```
// a generic function add()
```



```
int main() {
    int x[] = {1,2,3,4,5};
    double d[] = {1.2, 2.3, 3.4, 4.5, 5.6, 6.7};
```

```
    cout << "sum of x[] = " << _____ << endl; // Calculates the sum of 5 elements in x
    cout << "sum of d[] = " << _____ << endl; // Calculates the sum of 6 elements in d
```

```
sum of x[] = 15
sum of d[] = 23.7
```

Practice #4-1: Understanding Function Templates (2)

- Create a template function "reverseArray()" that reverses the order of elements in an array.
 - It receives two arguments (array, total number of elements).
 - Use the following code snippet:

```
int main() {  
    int x[] = {1, 10, 100, 5, 4};  
    reverseArray(x, 5);  
  
    for(int i=0; i<5; i++)  
        cout << x[i] << ' '; // Output should be: 4 5 100 10 1  
    cout << endl;  
}
```

Practice #4-2: Grade Processing Program #3

■ Problem:

- Modify the previously implemented Grade Processing Program #2.
- (File name: Practice#4_StudentID_Name.cpp)

Practice #4-2: Grade Processing Program#3 (1)

■ Encapsulation and Feature Enhancement

- Refactor GPA calculation, average GPA computation, and displaying all student grades into functions.
- Add a student search function: when a student's name is entered, display their student ID, name, and detailed grade information.
 - StdSearch(): Takes a student name and total number of students, returns a pointer to the student's data.
 - PrintOneData(): Receives a student pointer and prints the student's detailed information.

함수 이름	기능 요약	함수 선언
CalcGPA	과목 등급(Grade)을 평점(GPA)으로 환산하여 저장	<code>void CalcGPA(Subject &Sub);</code>
CalcAveGPA	과목들의 GPA 평균 계산	<code>float CalcAveGPA(Subject *Sub, int SubNum);</code>
PrintAllData	모든 학생의 성적 정보 출력	<code>void PrintAllData(const Student *pSt, int StudentNum);</code>
PrintOneData	하나의 학생에 대한 상세 성적 정보 출력	<code>void PrintOneData(const Student &rSt);</code>
StdSearch	학생 이름으로 검색 후 해당 학생 포인터 반환	<code>Student* StdSearch(Student *pSt, int StudentNum);</code>

Practice #4-2: Grade Processing Program#3 (2)

■ Inline Function Creation

- Move the menu output code into a function and implement it as an inline function.
- Implement input functions using overloading and inline:
 - InputValue(char*): for string input
 - InputValue(int&): for integer input

함수 이름	기능 요약	함수 선언
PrintMenu	메뉴 출력 (inline 함수)	<code>inline void PrintMenu();</code>
InputValue(char*)	문자열 입력 처리 (오버로딩, inline 함수)	<code>inline void InputValue(char *str);</code>
InputValue(int&)	정수 입력 처리 (오버로딩, inline 함수)	<code>inline void InputValue(int &i);</code>
PrintAllStdList	전체 학생 목록(이름, 학번만) 출력	<code>void PrintAllStdList(const Student *pSt, int StudentNum = 2);</code>
ModifyStdInfo	학생 이름으로 검색 후 이름과 학번 수정	<code>void ModifyStdInfo(Student *pSt, int StdNum);</code>

Practice #4-2: Grade Processing Program#3 (2)

■ Feature Addition

- Implement PrintAllStdList function:
 - Prints all stored students' names and student IDs.
 - Takes two parameters (structure, total number of students [default argument = 2]).
- Implement ModifyStdInfo:
 - Allows modification of a student's name and ID based on name search.

함수 이름	기능 요약	함수 선언
PrintMenu	메뉴 출력 (inline 함수)	<code>inline void PrintMenu();</code>
InputValue(char*)	문자열 입력 처리 (오버로딩, inline 함수)	<code>inline void InputValue(char *str);</code>
InputValue(int&)	정수 입력 처리 (오버로딩, inline 함수)	<code>inline void InputValue(int &i);</code>
PrintAllStdList	전체 학생 목록(이름, 학번만) 출력	<code>void PrintAllStdList(const Student *pSt, int StudentNum = 2);</code>
ModifyStdInfo	학생 이름으로 검색 후 이름과 학번 수정	<code>void ModifyStdInfo(Student *pSt, int StdNum);</code>

Practice #4-2: Sample Execution Output

■ 실행결과 예

학생 목록 보기 메뉴 선택 결과

```
===== 메뉴 =====
1. 학생 성적 입력
2. 전체 학생 성적 보기
3. 학생 이름 검색 보기
4. 전체 학생 목록 보기
5. 학생 정보 수정
6. 프로그램 종료
```

원하는 기능을 입력하세요 : 4

```
=====
학번      이름
=====
201450623  김홍도
20142034   김미소
20142056   손소라
=====
```

학생 정보 수정, 학생 이름 검색 메뉴 선택 결과

```
===== 메뉴 =====
1. 학생 성적 입력
2. 전체 학생 성적 보기
3. 학생 이름 검색 보기
4. 전체 학생 목록 보기
5. 학생 정보 수정
6. 프로그램 종료
```

원하는 기능을 입력하세요 : 5

검색 할 학생 이름 : 김홍도
* <김홍도, 201450623>의 정보를 수정하세요
이름 : 김홍진
학번 : 20145001

```
===== 메뉴 =====
1. 학생 성적 입력
2. 전체 학생 성적 보기
3. 학생 이름 검색 보기
4. 전체 학생 목록 보기
5. 학생 정보 수정
6. 프로그램 종료
```

원하는 기능을 입력하세요 : 3

검색 할 학생 이름 : 김홍진
이름 : 김홍진 학번 : 20145001

```
=====
과목명   과목학점   과목등급   과목평점
=====
영상처리       3           C        6.00
=====
                        평균평점      6.00
=====
```