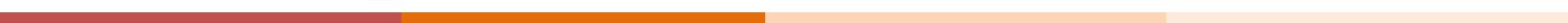


04. 개선된 함수 기능_실습

실습내용



■ 실습#4

- #4-1. 함수 템플릿 이해하기
- #4-2. 인라인함수 사용하여 성적처리프로그램#2 개선

실습#4-1 : 함수 템플릿 이해하기(1)

■ 배열과 크기를 매개 변수로 받는 add() 함수를 작성해보세요.

- add() 함수는 2개(배열, 크기)의 인자를 전달받는 제네릭 함수
- add() 함수에는 전달받은 배열의 모든 요소를 합하는 코드 작성
- add() 함수는 계산된 값을 반환

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

//제네릭 함수 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; // 배열 x와 원소 5개의 합을 계산
    cout << "sum of d[] = " << _____ << endl; // 배열 d와 원소 6개의 합을 계산
}
```

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

실습#4-1 : 함수 템플릿 이해하기(2)

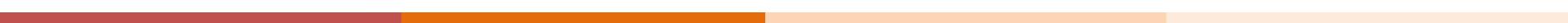
■ 배열의 요소 순서를 뒤집는 reverseArray() 함수를 템플릿으로 작성해 보세요.

- reverseArray()는 2개(배열, 배열요소의 총 개수)의 인자를 전달받는 제네릭 함수
- 다음의 코드를 활용하여 작성할 것

```
int main() {
    int x[] = {1, 10, 100, 5, 4};
    reverseArray(x, 5);

    for(int i=0; i<5; i++)
        cout << x[i] << ' '; // 4 5 100 10 1 이 출력된다.
    cout << endl;
}
```

실습#4-2 : 성적처리프로그램#3



■ 문제

- 이전에 작성된 성적처리프로그램#2를 수정한다.
(파일 명 : 실습과제#4_학번_홍길동.cpp).

실습#4-2 : 성적처리프로그램#3 (1)

■ 캡슐화 및 기능 추가

- 평점계산, GPA 평균 계산, 전체학생성적보기 기능들의 함수화
- 학생 검색 기능 추가(학생의 이름을 입력하면, 학번 이름 상세 성적 정보 출력)
 - StdSearch 함수: 학생의 이름과 저장된 총 학생의 수를 입력 받아 해당 학생의 정보가 저장된 공간의 포인터 반환
 - PrintOneData 함수: 학생 포인터를 입력받아 학생 상세 정보를 출력

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

실습#4-2 : 성적처리프로그램#3 (2)

■ inline 함수 생성 및 기능추가

- 메뉴출력코드 -> 함수 -> 인라인함수
- 문자열을 입력 받는 함수 `InputValue(char *)`, 정수를 입력 받는 함수 `InputValue(int &)` 구현
 - (오버로딩을 사용, inline 함수로 구현)
- 저장된 전체 학생들의 이름과 학번 출력 기능(`PrintAllStdList`): 함수호출 -> 학생들의 이름 학번 출력
 - 2개 (구조체, 총 학생수[디폴트인자사용=2]) 의 매개변수사용
- 학생정보 수정 기능(`ModifyStdInfo`): 학생이름검색 -> 이름과 학번 수정

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

실습#4-2 : 성적처리프로그램#3(2)

■ 실행결과 예

학생 목록 보기 메뉴 선택 결과

```
===== 메뉴 =====
1. 학생 성적 입력
2. 전체 학생 성적 보기
3. 학생 이름 검색
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
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
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
```