

□ 기본 개념 확인

(1) O, X로 답하세요.

- 1) C++는 C와 호환되지 않는다. ()
- 2) C++는 절차 지향 언어이다. ()
- 3) 실행 시간 효율성 저하를 최소화 하기 위하여 재귀 함수를 사용한다. ()
- 4) C++ 프로그램은 이미 컴파일 된 C언어의 목적 파일은 링크시켜 사용할 수 없다. ()
- 5) C++ 헤더 파일은 확장자를 생략한다. ()
- 6) main() 함수는 반드시 return문을 가지고 있어야 한다. ()
- 7) 이름 공간을 선언할 때 using 을 사용한다. ()
- 8) bool 타입의 변수에 저장된 값은 true 또는 false로 출력된다. ()
- 9) static_cast 연산자는 기본 자료형에 대하여 사용한다. ()
- 10) string 클래스를 사용하여 문자열을 처리할 때 이름공간은 생략해도 된다. ()

(2) 빈 괄호를 채워 넣으세요.

- 1) C++는 () 언어와의 호환성과 () 개념 추가를 목표로 설계되었다.
- 2) 작은 크기의 멤버 함수를 자주 호출할 때 발생하는 실행 시간 효율성 저하 문제는 ()함수로 해결한다
- 3) C++ 프로그램의 확장자는 () 이다.
- 4) () 헤더 파일은 표준 입출력을 위한 클래스와 객체, 변수 등이 선언되어 있다.
- 5) 이름 공간을 생략하려면 () 지시어를 사용한다.
- 6) bool 타입으로 선언된 변수에 저장된 값을 불 리터럴로 출력하려면 조작자 ()를 사용해야 한다.
- 7) C++ 에서 문자열을 처리하는 방법은 문자 배열 또는 () 클래스를 사용한다.
- 8) 공백이 포함된 문자열을 처리하려면 전역 함수 ()을 사용한다.
- 9) uniform initialization는 ()을 사용하여 변수, 배열, 객체 등을 초기화 한다.
- 10) 표준 입력 장치인 키보드로 데이터를 입력 받을 때 () 객체를 사용한다.

(3) 질문에 답하세요.

- 1) C++ 표준 라이브러리 그룹 3개를 설명하세요.
- 2) C++ 프로그램이 실행을 시작하는 함수 원형을 제시하세요.
- 3) C++ 표준에서 cin, cout 객체는 어떤 헤더 파일에 선언되어 있나요?
- 4) C++ 표준 라이브러리가 모두 선언된 이름 공간은 무엇인가요?
- 5) 이름공간을 선언할 때 사용하는 키워드는 무엇인가요?
- 6) 컴파일 때 변수에 초기화 된 값으로 형을 결정하는 자료형은 무엇인가요?
- 7) namespace hallym에 정의된 함수 software를 호출하는 문장을 쓰세요. 단, software 함수는 반환 값과 매

개변수를 가지지 않습니다.

- 8) 다음 문장에 오류가 발생하지 않도록 수정하세요. using 지시어는 사용하지 않습니다.

```
cout>>"C++";
```

- 9) 다음 중 컴파일 오류가 발생하는 문장은 무엇인가요?

① int d { 3.6 };

② int f = 3.6;

- (4) 다음 문장 중에서 틀린 부분을 올바르게 수정하세요.

① #include <iostream.h>

② using std namespace;

③ cin<<data;

④ 100>>cout;

- (5) 이름 공간을 지정하는 문장을 사용하지 않으려고 합니다. 아래의 코드를 수정하세요.

```
#include <iostream>
```

```
#include <string>
```

```
int main() {  
    string name;  
    cout << "이름을 입력하세요 : ";  
    cin >> name;  
    cout << "name : " << name << endl;  
    return 0;  
}
```

- (6) 제시된 프로그램에 대하여 질문에 답하세요.

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {  
    string name;  
    cout << "이름을 입력하세요 : ";  
    cin >> name;  
    cout << name << " 님 환영합니다 " << endl;  
    return 0;  
}
```

- 6-1) 다음과 같이 키보드로 "C++"을 입력하였을 때 결과를 제시하세요.

이름을 입력하세요? C++

6-2) 다음과 같이 키보드로 "C++ OOP"을 입력하였을 때 결과를 제시하세요.

이름을 입력하세요? C++ OOP

6-3) 공백이 포함된 문자열을 입력 받을 수 있도록 위의 프로그램을 수정 하세요. 단, 수정한 문장만 제시합니다.

(7) 컴파일 오류가 발생하지 않도록 빈칸을 채워 넣으세요.

```
#include <iostream>
```

```
int main()
{
    int count;
    cout << "반복 횟수를 입력하세요 : ";
    std::cin >> count;
    cout << count << " 회 반복합니다. "<<endl;
    return 0;
}
```

(8) 다음 프로그램을 cpp로 수정한 소스와 결과를 함께 제시하세요. auto변수로 변환 가능한 곳은 모두 변경하세요.

```
#include <stdio.h>
int sum(int s, int e); // 함수 원형 선언

int sum(int s, int e)
{
    int tmp, res = 0;

    if(s>e) {
        tmp = e;
        e = s;
        s = tmp;
    }
    for (int k = s; k <= e; k++)
    {
        res += k;
    }
    return res;
}
```

```

}
int main()
{
    int a, b;
    printf("첫 수와 마지막 수를 입력하세요 : ");
    scanf("%d %d", &a, &b);

    printf("%d부터 %d까지의 합은 %d 입니다\n", a, b, sum(a, b));

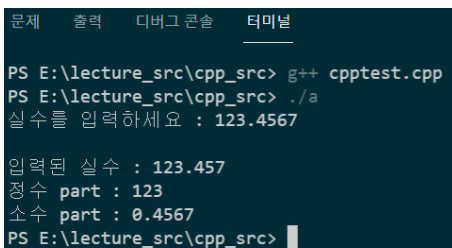
    return 0;
}

```

[프로그램 소스]

[실행 결과]

(9) 입력된 실수에서 정수 부분과 소수점 아래부분을 분리해서 출력하는 프로그램을 작성하세요



```

문제   출력   디버그 콘솔   터미널
PS E:\lecture_src\cpp_src> g++ cpptest.cpp
PS E:\lecture_src\cpp_src> ./a
실수를 입력하세요 : 123.4567

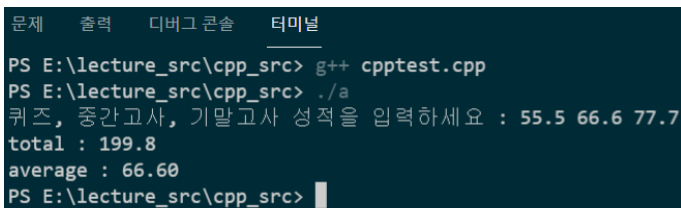
입력된 실수 : 123.457
정수 part : 123
소수 part : 0.4567
PS E:\lecture_src\cpp_src>

```

[프로그램 소스]

[실행 결과]

(10) 퀴즈, 중간고사, 기말고사의 성적을 사용자로부터 입력 받아 총합과 평균을 계산하는 프로그램을 작성하세요. 단, 이름공백은 선언하지 않습니다.



```

문제   출력   디버그 콘솔   터미널
PS E:\lecture_src\cpp_src> g++ cpptest.cpp
PS E:\lecture_src\cpp_src> ./a
퀴즈, 중간고사, 기말고사 성적을 입력하세요 : 55.5 66.6 77.7
total : 199.8
average : 66.60
PS E:\lecture_src\cpp_src>

```

[프로그램 소스]

[실행 결과]

(cpp_02_기본 슬라이드 마지막 페이지 실습 - 기본 입출력 활용) 입력한 10진 정수를 다양한 진법으로 출력하기 : 제어문 & 조작자 & 문자열 활용.

```
문제   출력   디버그 콘솔   터미널

PS E:\lecture_src\cpp_src> g++ cpptest.cpp
PS E:\lecture_src\cpp_src> ./a
10진수 입력 : 55

여러 진법으로 출력 하기
oct(8), hex(16), digit(10)
해당 진법 입력 : 16
=> 16진법 : 37
PS E:\lecture_src\cpp_src> ./a
10진수 입력 : 55

여러 진법으로 출력 하기
oct(8), hex(16), digit(10)
해당 진법 입력 : hex
=> 16진법 : 37
PS E:\lecture_src\cpp_src> █
```

[프로그램 소스]

[실행 결과]

•(cpp_02_기본 슬라이드 마지막 페이지 실습 - initializer_list 활용) 두번째 인수와 가장 가까운 거리에 있는 문자 출력하기.

```
int main() {
    cout << "{ 'd', 'p', 'r', 'w', 'g', 'f' }문자 중 h와 가까운 문자는 : ";
    cout << list_exam({ 'd', 'p', 'r', 'w', 'g', 'f' }, 'h') << endl;

    cout << "{ 'k', 'q', 'b', 'r', 'a', 'e', 'v', 'z'}문자 중 w와 가까운 문자는 : ";
    cout << list_exam({ 'k', 'q', 'b', 'r', 'a', 'e', 'v', 'z'}, 'w') << endl;
}
```

```
{ 'd', 'p', 'r', 'w', 'g', 'f' }문자 중 h와 가까운 문자는 : g
{ 'k', 'q', 'b', 'r', 'a', 'e', 'v', 'z'}문자 중 w와 가까운 문자는 : v
```

[프로그램 소스]

[실행 결과]
