

Discussion Questions

1. How does having a widely adopted C++ standard help game programmers?

C++가 **standard**로서 널리 채택된다면, 게임 프로그래머들이 받을 수 있는 여러 이점이 있다. 그중 세 가지를 골라 설명하자면 호환성, 이식성, 성능이 있다.

첫째, 여러 곳에서 사용되는 C++을 채택하는 경우, 오랜 시간이 지나 과거에 작성한 코드가 낡거나 새로운 기술이 생기더라도 서로 호환되지 못하는 위험 부담이 줄어든다.

둘째, 표준화가 이루어진 언어를 채택하는 경우에는 C++로 작성한 코드를 여러 플랫폼에 이식할 수 있어 코드를 여러 번 작성해야 하는 번거로움을 덜 수 있다.

셋째, 성능과 효율성이 좋기로 알려진 C++은 특히 게임 개발에 유리하기 때문에 비교적 안정적이고 빠른 게임 개발 및 실행을 이룰 수 있게 한다.

3. Why might you define a new name for an existing type?

기존 **type**에 새 이름을 정의하는 이유는 코드의 가독성을 향상하기 위함이다. 새 이름을 사용하면 코드에서 사용되는 **type**를 더욱 명확하게 표현할 수 있다. 더불어 프로젝트나 코드베이스에서 일관된 이름을 유지하면 코드의 의도 유지 및 관리에 용이하다.

Exercises

2. What's displayed by each line in the following code snippet? Explain each result.

```
cout << "Seven divided by three is " << 7 / 3 << endl;  
cout << "Seven divided by three is " << 7.0 / 3 << endl;  
cout << "Seven divided by three is " << 7.0 / 3.0 << endl;
```

1) `cout << "Seven divided by three is " << 7 / 3 << endl;`

Result: Seven divided by three is 2

문자열 "Seven divided by three is "를 출력하고, 이어서 '7 / 3'의 결과를 출력한다. 정수의 나눗셈이므로 소수 부분은 출력하지 않기 때문에 '2'의 값이 도출된다.

2) `cout << "Seven divided by three is " << 7.0 / 3 << endl;`

Result: Seven divided by three is 2.33333

문자열 "Seven divided by three is "를 출력하고, 이어서 '7.0 / 3'의 결과를 출력한다. 실수와 정수의 나눗셈이므로 연산값은 실수가 되어 '2.33333'의 값이 도출된다.

3) `cout << "Seven divided by three is " << 7.0 / 3.0 << endl;`

Result: Seven divided by three is 2.33333

문자열 "Seven divided by three is "를 출력하고, 이어서 '7.0 / 3.0'의 결과를 출력한다. 실수의 나눗셈이므로 연산값은 실수가 되어 '2.33333'의 값이 도출된다.