**(1) Orthogonality가 C언어에서 발생시킬 수 있는 오류의 예를 들어라!**

: Fewer exceptions

**(2) C++에서 최근에 도입된 개념인 Lambda expression, smart pointer, uniform initializer, variadic template를 조사하라.**

1) lambda expression : 함수 객체를 사용하는데 불편함을 느껴서 만들어지게 되었다.

함수 객체에 비해 코딩이 간편해지고, 코드의 가독성이 향상된다.

람다(lambda)의 사용법은 간단하다.

- []{}

- [](인자){}

람다 소개자 : [] (Lambda Introducer)

파라미터 지정자 : () (Parameter Specifier)

람다 몸체 : {} (Lambda Body)

이 두가지 방법중 하나를 사용하면 되는데, 람다 표현식에 변수를 넘기고 싶은 경우에는 가운데 괄호를 넣고 인자를 넣으면 된다.

람다 함수는 람다 소개자로 시작해야 인식을 하는데, 람다 소개자라는 것은 바로 맨 처음 대괄호를 지칭한고 파라미터 지정자는 람다 함수에 넘겨줄 인자를 입력할 수 있는 메서드다. 그리고 람다 몸체는 일반 함수와 쓰임새가 같으며, {} 내부에 람다 함수가 수행할 일을 작성한다.

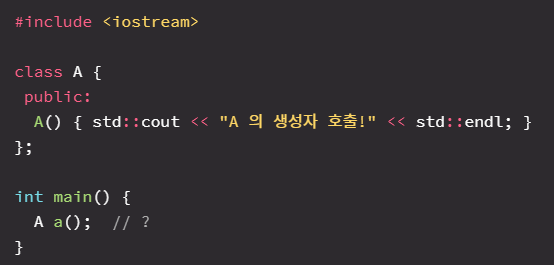
맨 뒤에 ()를 붙이고 인자를 넣어주면 함수가 시작이 된다.

2) smart pointer : 메모리 누수(memory leak)로부터 프로그램의 안정성을 보장하기 위해서 만들어졌다.

smart pointer는 포인터처럼 동작하는 클래스 템플릿인데. 사용이 끝난 메모리를 자동으로 해제해준다.

new를 이용해 포인터가 실제 메모리를 가리키도록 초기화한 뒤에 기본 포인터를 스마트 포인터에 대입해서 사용한다. 스마트 포인터가 사용이 끝나면 소멸자가 delete 키워드를 사용하여 할당된 메모리를 자동으로 해제해준다.

3) uniform initializer : 원치 않는 타입캐스팅으로부터의 오류와 ()의 사용으로 인한 불편함을 방지하기 위해서 만들어지게 되었다.



위와 같은 코드는 실행하면 아무것도 출력되지 않는데, 그 이유는 위 코드가 A의 객체 a를 만든 것이 아니라 ㄴA를 리턴하고 인자를 받지 않는 함수 a를 정의한 것이기 때문이다.

위와 같은 문제는 ()가 함수의 인자들을 정의하는데도 사용되고, 일반적인 객체의 생성자를 호출하는데도 사용되기 때문에 발생한다.

이 문제를 방지하기 위해서 uniform initializer를 사용하는데 위의 ()를 {}로 바꿔주면 된다. ()를 {}로 바꿔주게 되면 Implicit type conversion을 불허하게 변환된다. 이렇게 Implicit type conversion을 불허하여 원하지 않는 type casting 을 방지하게 되어 오류를 방지할 수 있다.

4) variadic template :

template parameter pack, function parameter pack을 이용하여 받은 인자만큼에 해당하는 인자를 가진함수를 호출하여 사용할 수 있다. 유의할 점은 C++은 컴파일 시에 앞서 정의된 함수들만 사용할 수 있으므로, 순서에 유의해야 한다.

variadic template을 잘 활용하면 작성해야 하는 코드의 양을 줄일 수 있다.

(3) **스마트폰이 일반화하면서 ‘Platform’이란 말이 일반화하고 있다. ‘Platform’이란 무엇이며, 일반 응용프로그램과 어떤 차이가 나는가? 현재 널리 쓰이는 smartphone platform을 조사하고, 이 platform을 만드는 데 사용한 프로그래밍언어와 응용소프트웨어(앱)과 연결 가능한 언어 및 연결 방법을 조사하라.**

1) 플랫폼(Platform)이란?

: 플랫폼이란 여러 사람들이 쉽고 편리하게 사용할 수 있다는 뜻이 담겨있다.

플랫폼은 각 분야(하드웨어, 소프트웨어, 서비스 등)에서 다양하게 사용된다.

예를 들자면,

하드웨어 플랫폼은 표준 공정을 통해 다양한 제품을 만들어내는 기반과 도구를 뜻한다.

소프트웨어 플랫폼은 여러 기능을 제공해주는 공통 실행환경을 뜻한다.

서비스 플랫폼은 다른 서비스들이 내 서비스를 쉽게 활용할 수 있게 해주는 인터넷 기반의 기술 환경을 뜻한다.

이렇게 플랫폼이라는 이름이 붙으면 해당 플랫폼을 이용해서 사용자 및 관리자, 구매자 및 판매자가 편리하게 사용할 수 있는 것이라는 뜻으로 해석할 수 있다.

2) 플랫폼(Platform)과 일반응용프로그램의 차이점?

: 컴퓨터와 관련해서 Platform 은 응용프로그램이 실행될 수 있는 기초를 이루는 컴퓨터 시스템을 의미한다. 예를 들면 PC에서는 윈도우 XP, 맥킨토시 등을 들 수 있고, 그 기반위에서 응용프로그램이 동작하게 된다.

3) 현재 널리 쓰이는 smartphone platform?

: 현재 널리 쓰이는 smartphone platform은 크게 두 가지로 Android와 ios를 들 수 있다.

4) 이 플랫폼을 만드는데 사용한 프로그래밍 언어?

Android는 운영체제는 대부분 C로 만들어졌다.(리눅스 기반)

ios의 운영체제 또한 대부분 C로 만들어졌다.(유닉스 기반)

5) 응용소프트웨어와 연결언어, 연결방법

Android의 응용소프트웨어는 Java, Kotlin으로 만들어진다. 안드로이드는 기존의 자바 가상 머신(JVM : Java Virtual Machine)과 다른 가상 머신인 안드로이드 런타임을 통해 자바와 코들린으로 작성된 응용프로그램을 별도의 프로세스에서 실행하는 구조로 되어 있다.

ios의 응용소프트웨어는 Objective-C, Swift 등으로 만들어진다.

**(4) 결과 출력과 해석:**

**for( int i: {2, 3, 5, 7, 9}) { std::cout << i << std::endl}**

출력

2

3

5

7

9

해석 : i에 2라는 인자를 받아서 i를 std 표준라이브러리를 나타내는 std:: 접두어를 이용해 콘솔에 출력하고, 개행 문자 출력,

이 과정을 9라는 인자를 받을때까지 반복