

Quiz #1 Programming Language, 2017, Term 2

Name () St. Id ()

NB) 배점동일, 부분점수 인정

1. 프로그래밍 언어의 구성 요소 4가지를 쓰시오.

구문구조 (Syntax), 이름(Name), 타입(Type), 의미구조(Semantics)

2. 성공적인 프로그래밍 언어가 가지는 주요 특징들을 열거하시오.

단순성과 가독성, 묶기binding의 명확성, 신뢰성, 지원성, 추상화
Abstraction, 직교성Orthogonality, 구현의 효율성

3. 추상화와 직교성에 대해 설명하시오.

<추상화>

데이터

- 프로그래머 정의 타입 및 클래스
- 클래스 라이브러리

프로시저

- 프로그래머 정의 함수
- 표준 함수 라이브러리

<직교성>

상호 독립된 소규모 기본 연산만으로 프로그램을 구축할 수 있는 언어를 직교적(orthogonal)이라고 한다.

- 필요한 예외적 규칙의 최소화 = 개념적으로 단순

- ✓ 예: 함수 인수의 타입을 제한함

- 효율성과 타협의 여지

- ✓ 직교성이 좋아지면 효율성이 낮아질 수 있고,

- ✓ 효율성이 좋아지면 직교성이 낮아질 수 있다.

4. 직교성을 나타내는 C, C++, java의 기능을 찾으시오. 교제에서 설명한 것이 아닌, 반직교성을 나타내는 기능을 찾으시오.

In Java, the (overloaded) operator + denotes string concatenation as well as addition is an instance of orthogonality. Operator overloading, however, is not a core feature of Java, as it is in Ada. Other Java operators are generally not overloaded or overloadable.

Non-orthogonality is widespread in C, C++, and Java. In C and C++, for instance, string concatenation is done using the strcat function. However, in C++ the display of two strings s and t is written as cout << s << t.

5. 표준 문서에서 특정 컴파일러가 지원하는 언어 기능에 대해서 설명할 때, 불일치(nonconformance)란 무엇을 의미하는가?

A C++ platform (compiler) is nonconformant if it supports features in a way that is incompatible with the C++ standard. On the other hand,

language extensions are features supported by a platform that exceed those defined by the standard (but do not make the compiler nonconformant).

Many C++ compilers support extensions to the C++ standard, but are otherwise conformant with it. However, programs that utilize the extensions supported by one compiler may not port to a different platform.

More information on C++ standardization can be found at

<http://www.open-std.org/jtc1/sc22/wg21/> and

<http://www.research.att.com/~bs/C++.html>

<End of the Quiz 1>