**Test #9. 2013.5.7 프로그래밍언어론 02 분반**

학번:

이름:

1. 다음 각 항목이 참이면 O, 아니면 X 를 하시오.
   * 1. 수학적 함수의 참조 투명성(referential transparency)에 대응되는 프로그래밍언어의 개념은 side-effect를 유발하지 않는다는 것이다.
     2. 사용자 정의 overloaded operator의 우선순위(precedence), 결합법칙(associative rule) 등은 원래의 연산자와 동일하다.
     3. Java의 generic 함수는 수행되는 코드 크기가, 동일한 기능을 하는 C++의 template 함수보다 일반적으로 더 크다.
     4. Parameter의 타입이 모두 동일하고 리턴 타입으로만 구별될 수 있는 Overloaded subprogram들이 정의되어 있을 때, C++은 오류를 내지만 Java는 모두 수행된다.
     5. C#에서는 int myInt = MyClass.foo<int>(37); 대신에 int myInt = MyClass.foo(37); 로 쓸 수 있다.
2. 다음이 설명하고 있는 용어는? ( )

“변수에 저장할 수 있고, Subprogram에게 parameter로 전달할 수 있고, Function의 결과로 return이 가능하다.”

3. C++에서 유사한 기능을 하는 다음 (1), (2) 두 코드의 장단점을 비교하시오.

(1)

template <class Gtype> // Gtype: generic type

Gtype max(Gtype a, Gtype b) {

return a > b ? a : b;

}

(2)

#define max(a, b) ((a) > (b)) ? (a) : (b)

답:

1. 다음 각 항목이 참이면 O, 아니면 X 를 하시오.
   * 1. 수학적 함수의 참조 투명성(referential transparency)에 대응되는 프로그래밍언어의 개념은 side-effect를 유발하지 않는다는 것이다. O
     2. 사용자 정의 overloaded operator의 우선순위(precedence), 결합법칙(associative rule) 등은 원래의 연산자와 동일하다. O
     3. Java의 generic 함수는 수행되는 코드 크기가, 동일한 기능을 하는 C++의 template 함수보다 일반적으로 더 크다. X
     4. Parameter의 타입이 모두 동일하고 리턴 타입으로만 구별될 수 있는 Overloaded subprogram들이 정의되어 있을 때, C++은 오류를 내지만 Java는 모두 수행된다. X
     5. C#에서는 int myInt = MyClass.foo<int>(37); 대신에 int myInt = MyClass.foo(37); 로 쓸 수 있다. O
2. 다음이 설명하고 있는 용어는? ( first class object )

“변수에 저장할 수 있고, Subprogram에게 parameter로 전달할 수 있고, Function의 결과로 return이 가능하다.”

🡺first order object도 가능

3. C++에서 유사한 기능을 하는 다음 (1), (2) 두 코드의 장단점을 비교하시오.

(1)

template <class Gtype> // Gtype: generic type

Gtype max(Gtype a, Gtype b) {

return a > b ? a : b;

}

(2)

#define max(a, b) ((a) > (b)) ? (a) : (b)

(2) 단점 이자 (1)장점

타입검사를 하지 않는다.

actual parameter가 side effect를 가지면 결과 예측이 어렵다.

1. 단점이자 (2)장점

정의가 간단하다. (이건 거의 안 써도 맞게 하자)