1. 다음 문제를 읽고 맞으면 True에 틀리면 False를 선택하시오. 답을 표시하고 맞췄을 경우에는 +2, 틀렸을 경우에는 -1점으로 계산함. 정답을 표시하지 않고 틀렸을 경우에는 0점 처리함 (총 20점)

|  |  |  |  |
| --- | --- | --- | --- |
| 1) | Java 언어는 사용자 정의 타입에 대해서 Reference model of variables를 사용한다. | True | False |
| 2) | C/C++언어에서 expression이 l-value로 사용될 수도 있다. | True | False |
| 3) | Java에서 built-in (primitive) 타입을 결과 값으로 생성하는 expression은 l-value로 사용할 수 있다. | True | False |
| 4) | C언어에서는 안되지만, C++에서는 구조체에 대한 레퍼런스를 함수에서 반환할 수 있다. | True | False |
| 5) | Expression을 evaluation하는 순서는 side effect와 코드를 빨리 컴파일하는 문제 때문에 중요하게 고려된다. | True | False |
| 6) | C/C++/Java언어의 for loop는 enumeration controlled loop의 전형적인 예를 보여준다. | True | False |
| 7) | 모든 Recursion코드는 Iterative code로 변환 가능하다. | True | False |
| 8) | inline함수는 macro함수보다 처리 속도가 빨라서 사용한다 | True | False |
| 9) | 객체지향언어에서 destructor가 제공되는 주된 경우는 garbage collection이 지원되지 않을 때이다. | True | False |
| 10) | 전달되는 인자의 크기가 큰 경우에는 call by reference를 사용하는 것보다 call by value로 인자를 전달하는 것이 더 효율적이다. | True | False |

1. 다음 문제를 읽고 맞으면 True에 틀리면 False를 선택하시오. 답을 표시하고 맞췄을 경우에는 +2, 답을 표시하고 틀렸을 경우에는 -1점, 정답을 표시하지 않으면 0점 처리함 (총 20점)

|  |  |  |  |
| --- | --- | --- | --- |
| 1) | Java 언어는 사용자 정의 타입에 대해서 Value model of variables를 사용한다. | True | False |
| 2) | 모든 iterative code는 recursion으로 변환 가능하다. | True | False |
| 3) | Java에서 built-in (primitive) 타입을 결과 값으로 생성하는 expression은 l-value로 사용할 수 없다. | True | False |
| 4) | C/C++언어에서는 구조체에 대한 레퍼런스를 함수에서 반환할 수 없다. | True | False |
| 5) | Expression을 evaluation하는 순서는 side effect와 코드의 최적화 문제때문에 중요하게 고려된다. | True | False |
| 6) | C/C++/Java언어의 for loop는 enumeration controlled loop보다는 logically controlled loop에 좀 더 가까운 모습을 보인다. | True | False |
| 7) | C/C++언어에서 expression이 l-value로 사용될 수도 없다. | True | False |
| 8) | inline함수는 일반적으로 macro함수처럼 처리 속도가 빠르면서도 scope나 abstraction을지원하는 장점이 있다. | True | False |
| 9) | 객체지향언어에서 garbage collection이 지원되면 destructor가 제공되지 않아도 된다. | True | False |
| 10) | 전달되는 인자의 종류에 따라서 call by value보다는 call by reference를 사용해서 인자를 전달하는 것이 더 효율적이다. | True | False |



1. 다음 문제를 읽고 맞으면 True에 틀리면 False를 선택하시오. 답을 표시하고 맞췄을 경우에는 +2, 틀렸을 경우에는 –2점으로 계산함. 정답을 표시하지 않고 틀렸을 경우에는 0점 처리함 (총 20점)

|  |  |  |  |
| --- | --- | --- | --- |
| 1) | 컴파일러란 한 개의 프로그래밍 언어를 target machine code로 변환하는 프로그램을 의미함 | True | False |
| 2) | 인터프리터 언어는 Dynamic Scope를 사용함 | True | False |
| 3) | alias란 한 개의 object에 두 개 이상의 이름이 붙는 경우를 의미함 | True | False |
| 4) | Polymorphism이란 한 개의 이름으로 여러 개의 object를 사용하는 것을 의미함 | True | False |
| 5) | coercion이란 자동 형 변환 (automatic type conversion)을 의미함 | True | False |
| 6) | 아래 코드는 C# 코드는 유효하지 않음  class A {  const int N = 10;  void foo() {  const int M = N;  const int N = 20;  ... | True | False |
| 7) | C언어에서는 블록 기호({ ... })와 함께 새로운 Scope를 생성 | True | False |
| 8) | 다음 파이썬 코드를 실행시키면 화면에 출력되는 코드는 70이다  x = 90  def f1():  x = 70  def f2():  print(x)  f1()  f2() | True | False |
| 9) | C언어에서 static 변수는 global life와 local binding을 유지함 | True | False |
| 10) | 아래 코드에서 f1함수 body에서의 Referencing Environment는 f1함수의 a, f2 함수, 전역변수 b이다.  void f2(int b);  int a, b;  void f1(int a)  { ... }  void f2(int b)  { ... } | True | False |