2018 년 2 학기 프로그래밍언어론 기말고사

학번:_____ 이름:_____

1. Type equivalence 과 type compatibility 의 차이에 대해서 간략하게 설명한다 (10 점)

2. 아래 코드에서 Deep Binding 과 Shallow Binding 이 사용될 때 출력되는 결과를 보인다. (20 점)

```
void (*S)(int n);
void (*P)();

int x;

void setX(int n) {
    x = n;
}

void printX() {
    printf("%d", x);
}
```

```
void foo(S s, P p, int n) {
    int x = 5;
    if (n == 1 || n == 3)
        setX(n);
    else
        s(n);
    if (n == 1 || n == 2)
        printX();
    else
        p();
}
```

```
setX(0);
foo(setX, printX, 1);
printX();
setX(0);
foo(setX, printX, 2);
printX();
setX(0);
foo(setX, printX, 3);
printX();
setX(0);
foo(setX, printX, 4);
printX();
```

Shallow binding	Deep binding

3. Short-circuit evaluation 이 효율적이거나 유용한 경우를 보이는 코드의 예를 들고(본인이 선호하는 언어의 문법을 사용), 그 이유를 설명하라 (15 점)

4. 최근에 설계되는 언어들이 iterator 개념을 언어에 직접 넣거나 객체형태의 iterator 를 지원하는 경우가 많다. 기존 반복문에 비해 iterator 의 장점은 무엇인가에 대해 간략하게 설명한다 (10 점).

5. Value model of variables 를 따르는 C 언어로 아래 코드를 작성했다. (1)은 deep 또는 shallow assignment 인지 답하고, 아래 코드에서 나타날 수 있는 문제와 해결 방법이 무엇인지 간략하게 설명하라 (10 점)

```
typedef struct {
    int x, y;
    char* data
} A;
A a = { 0, 0, NULL };
a.data = (char*) malloc(100);
strcpy(a.data, “hello”); /* 문자열 복사 */
A b = a;                  /* (1)  */
free(a.data);
printf(“x = %d, y = %d, data = %s\n”, b.x, b.y, b.data);
```

6. 오른쪽 코드를 보고 applicative order 와 normal order 방법으로 인자가 전달되는 방법 중에서 어떤 것이 더 효율적인지 명시하고 그 이유를 설명하라 (10 점)

```
void f(int a) {
    int b = a * a * 4;
    …
}

int main(void) {
    f(g(3));
}
```

7. 일반 recursion 에 비해 tail-recursion 이 효율적일 수 있는 이유를 간략하게 설명한다 (10 점)

8. 왼쪽 코드는 name equivalence 를 지원하는 언어로 작성되었다. strict name equivalence 와 loose name equivalence 가 지원되는 경우에, 어떤 변수들이 equivalent 한지 보인다 (15 점).

```
typedef struct {
    int a, b;
} A;
typedef struct {
    int c;
    int d;
} B;
typedef A NEWA;
```

```
struct C {
    int c, d;
};

A a, b;
B c, d;
A e;
NEWA f;
struct C g;
```