

1. 부 프로그램 (15점)

- (a) “계산”을 파라미터로 넘겨서 ascending 및 descending sort를 할 수 있는 bubble sort 서브프로그램 “HighOrder_Sort(OP, int Arr[], size)”를 c-like한 문법으로 작성하고, 이 서브프로그램을 ascending 및 descending sort하기 위하여 어떻게 호출하는지 호출하는 statement를 작성하라. (10점)
- (c) (b)에서 작성한 부프로그램이 generic sorting할 수 있게 하기 위하여서는 compile time 혹은 run-time에 어떤 일을 하여야 하는지 설명하라. (5점)

2. 위와 같은 Pascal-like한 프로그램에서 아래와 같은 5가지 parameter-passing 방법들에 의하여 P()가 호출된다고 가정하였을 때, 이 프로그램의 수행이 끝난뒤 v[3]과 v[4]에 저장된 내용은 각각 무엇인가? 단, by result 및 by copy인 경우에 actual parameter의 주소는 호출 전에 계산된다고 가정하고, 최종 값을 알 수 없는 경우는 Unknown으로 답할 것. (4점 x 5 = 20점)

```
begin
  integer k
  integer array v[10]
  procedure P(a)
    integer a
    begin
      a := a + 1
      v[k] := a + 1
      k := 3
      a := a + 2
    end
  v[3] := 6
  v[4] := 8
  k := 4
  call P(v[k])
end
```

- (a) by value (4점)
- (b) by result (4점)
- (c) by copy (4점)
- (d) by reference (4점)
- (e) by name (4점)

3. FORTRAN77은 parameter-passing 방법으로 pass-by-reference을 사용하지만, 다음과 같은 형태의 호출도 가능하다. 아래의 물음에 답하라. (10점)

```
CALL P(X, X+Y, 2)
```

- (a) 이와 같은 호출을 FORTRAN77에서는 어떻게 구현하였는지 설명하라. (5점)
- (b) 이런 mechanism을 이용하여 variable X를 FORTRAN77에서 pass-by-value 형태로 passing할 수 있는 방법을 예를 들어 설명하라. (5점)

4. Subprogram (24점)

- (a) Overload Subprogram 기능을 제공하는 언어에서 함수 호출 (call)에 대한 함수 정의 (definition) binding은 무엇에 의하여 결정되는지 설명하라. (5점)
- (b) C++ 언어의 in-line 함수의 장점을 macro와 subprogram과 비교하여 설명하라. (5점)
- (c) 다음 각각 parameter passing 방법에서 actual parameter에 대한 주소 binding time이 언제인지 구체적으로 설명하라. (각 3점 x 3 = 9점)
(가) call by copy (나) call by reference (다) call by name
- (d) Ada 언어는 function 의 formal parameter는 항상 in-mode로 선언되어야 한다. 그 이유를 설명하라. (5점)

5. 아래와 같은 C++ 프로그램에 대하여 답하라. (10점)

- (a) error가 발생한 곳은 어디인가? (5점)
- (b) 이런 error는 언제 어떤 이유 때문에 발생하는가? (5점)

```
template <class T>
void swap (T& x, T& y) {
  T temp;
  temp = x ;
  x = y ;
  y = temp ;
}

main() {
  int i, j ;
  char str1[100], str2[100], ch;
  float m, n ;
  swap(i, j);
  swap(m, n);
  swap(str1[50], str2[33]);
  swap(i, ch);
  swap(str1, str2);
}
```

6. 다음의 Algol-like한 언어로 작성된 프로그램에 대하여 다음 물음에 답하여라. 단, parameter passing 방법은 call-by-reference를 사용한다고 가정한다. (15점)

- (a) 프로그램의 수행이 화살표 (가)의 문장을 수행한 후에 stack에 있는 activation record instance들의 내용을 각 local 변수, formal parameter의 값, static chain과 dynamic chain을 중심으로 나타내어라. (5점)
- (b) 프로그램의 수행이 화살표 (나)의 문장을 수행한 후에 stack에 있는 activation record instance들의 내용을 각 local 변수, formal parameter의 값, display와 dynamic chain을 중심으로 나타내어라. (5점)
- (c) (가)의 싹점에서 BIGSUB에서 선언한 XXX 변수를 참조하는 문장을 static chaining 방법을 이용하여 구현할 때와 display를 이용하여 구현할 때 각각에 대하여, compiler에 의하여 만들어지는 코드 형태와 수행할 때 일어나는 동작들에 대하여 설명하라. (5점)

```
procedure BIGSUB ;
var XXX : integer ;
var YYY : boolean ;
procedure C ; forward ;
procedure A (boolean flag) ;
  procedure B ;
  begin {of procedure B}
    A(false) ;
  end ; {of procedure B}
begin {of procedure A}
  if flag then B ;
  else C ;
end ; {of procedure A}

procedure C ;
  procedure D(integer YYY) ;
  begin {of procedure D}
    XXX = XXX + 1 ;
  end ; {of procedure D}
  begin {of procedure C}
    D(XXX) ;
  end ; {of procedure C}

begin {of procedure BIGSUB}
  YYY = true ;
  XXX = 5 ;
  A(YYY) ;
end ; {of procedure BIGSUB}
```

(가) →

(나) →

7. Python 프로그래밍 (10점)

리스트 voca에는 몇 개의 단어가 저장되어 있다 (중복된 단어가 있다). 각 단어가 voca에 몇 번 저장되어 있는지를 출력하는 프로그램을 작성하시오. (반드시 사전을 이용하여야 한다) voca의 내용이 바뀌면 결과도 따라서 바뀌어야 한다 (아래 예제 참고).

출력 결과

```
voca = ['python', 'c', 'c++', 'java',
        'java', 'javascript', 'python', 'ruby',
        'swift', 'c', 'c', 'java', 'python',
        'ruby', 'swift', 'cobol', 'cobol',
        'r', 'r', 'c', 'c', 'r', 'html', 'html',
        'ruby']
```

```
voca_dict = {}
```

```
for word in voca:
    /* (a). 여기를 채워라 (5점) */
```

```
for /* (b). 여기를 채워라 (5점) */
    print(w, '-', c, 'times')
```

```
python - 3 times
swift - 2 times
java - 3 times
html - 2 times
r - 3 times
c - 5 times
ruby - 3 times
c++ - 1 times
javascript - 1 times
cobol - 2 times
```

8. 입력된 리스트의 첫 번째 데이터와 마지막 데이터가 같은지 판단하는 프로그램을 가정하자. (25점)

(a) 아래 결과를 내는 Python 프로그램 함수 check(X)를 정의하여라. Recursion을 사용하지 말 것. (5점)

```
def check(L):
    /* 여기를 채워라, */

# main
A = [3,5,4,6,7]
B = [1,2,3,4,5]
C = [1,5,4,7,1,7,8,9,1]
print(check(A))    /* → False */
print(check(B))    /* → False */
print(check(C))    /* → True */
```

(b) 아래 결과를 내는 Prolog 프로그램 check(X)를 정의하여라. (10점)

```
:- check([3,5,4,6,7]) → Fail
:- check([1,2,3,4,5]) → Fail
:- check([1,5,4,7,1,7,8,9,1]) → True
```

(c) 위에서 작성한 Python 프로그램과 Prolog 프로그램의 길이가 L인 리스트를 check하는 경우의 각 프로그램의 평균 시간 복잡도 (Time Complexity)를 계산하라. (10점)

python : $O(1)$
prolog : list 개수만큼 recursion $O(n)$

9. 아래 Prolog 예제 프로그램 및 질의 예제를 이용하여 아래 물음에 답하라. (20점)

```
1. append([], L, L).
2. append([X|L1], L2, [X|L3]) :- append(L1, L2, L3).
   /* :- append([1,2,3,4], [5,6,7], Y). */
   /* :- append([a,b,c,d], [e,f,g,h], Y). */
```

```
member(X, [X|_]).
member(X, [_|Y]) :- member(X,Y).
/* :- member(1, [a,b,c,d]). */
```

원래는 넘어오는 타입을 선언하지 않지만
넘어갈 때(unification) 타입 바인딩(dynamic)

(a) Prolog 프로그램의 Unknown Value를 나타내는 변수의 타입 바인딩은 언제 일어나는가 ?
Static/Dynamic 중에서 고르고, 그 중에서 특정 시간을 지정하여 설명하라. (5점)

generic: 여러종류의 데이터타입을 처리하는 함수

(b) Prolog 언어는 generic function (혹은 Rule) 기능을 제공하는가 ?
그 이유를 위의 append 질의 예제를 이용하여 설명하라. (5점)

overloaded subprogram이고 generic이다

(c) Prolog 언어의 Scoping Rule은 무엇인가? ① Static 혹은 ② Dynamic 혹은 ③ "Scoping Rule이 없다" 중에서 어떤 것인가? 이유와 함께 설명하라. (5점)

scoping rule 없다

(d) Prolog는 overloaded function과 비슷한 기능을 제공한다. C++ 언어에서는 overload function을 구별하는 방법은 컴파일시에 함수의 파라미터 타입을 비교하여 결정한다. Prolog에서 같은 이름을 가지고 있는 overloaded fact 혹은 rule을 구별하는 방법에 대하여 설명하라. (5점)

매개 변수 리스트가 매치되는 걸(unification) 선택체

1. 다음의 용어들을 간단히 설명하라. (각 3점 x 5 = 15점)

- (a) FORTRAN 언어에서 Non-local Reference를 제공하는 방법 common을 사용해 같은 곳에 저장
- (b) static chain과 display는 궁극적으로 무엇을 구현하기 위한 방법인지 설명하고, non local reference를 처리 위해
C 언어는 이런 구현이 필요 없는 이유를 설명하라. 변수, 리턴을 미리 선언하는 것, subprogram이 nesting안되기 때문에
- (c) C언어에서 function prototyping을 하는 이유 actual, formal param의 타입체크, local 아닌 global이기 때문
- (d) 병행처리 (Concurrent Processing)과 병렬 처리 (Parallel Processing)을 비교 설명
- (e) 만약 object-based 언어에서 class 기능이 없이 같은 특성을 갖는 여러 개의 객체를
생성하려면 어떻게 프로그래밍하여야 하는지 설명 여러 프로그램을 나눠서 동시에 실행하는 것처럼 보이게 함

객체 프로그래밍 기능 : object를 틀을 만들어 만든다 - class

object based - class가 없는 객체 프로그래밍. 각 instance마다 같은 구조로 만들어야함

2. Parameters That are Subprogram Names (15점)

- (a) 함수 이름과 구간 (low, high)을 파라미터로 받아서, 적분하는 함수를 작성하고 (5점),
이 함수를 호출하는 예제를 C-like 언어로 작성 (5점)하라.
- (b) 이 함수가 generic subprogram이 되기 위하여서는 어떤 부분이 어떻게 바뀌어야 하는가?
언어 syntax에서 변경되어야 하는 부분과 수행 mechanism에서 변경되어야 하는 부분을
각각 설명하라. (5점)

타입을 런타임 때 다이나믹하게 바인딩 해서 맞는 거 수행

3. 다음과 같은 C++ 프로그램에 대하여 아래의 물음에 답하라. (10점)

```
class A {
private:
    int a;
protected:
    int b;
public:
    A() {
        b = a = 3;
        printf("Creating A\n");
    }
    ~A() {
        printf("Destroying A !\n");
    }
    sub1() {
        printf("a=%d\n", a);
    }
};

class B : public A {
private:
    int c;
public:
    B() {
        c = 4;
        c = a;
        printf("Creating B !\n");
    }
    ~B() {
        printf("Destroying B !\n");
    }
    sub1() {
        A::sub1();
        printf("b=%d, c=%d\n", b, c);
    }
};
```

```
main() {
    B b;
    b.sub1();
}
```

- (a) 이 프로그램은 error가 있다. 어느 문장이 왜 error인지 설명하라. (5점)
- (b) 위의 error 문장을 comment처리하고 난 후에 이 프로그램에 의하여
print되는 결과를 써라. (5점)

4. 다음과 같은 C++ 프로그램의 print되는 결과는 무엇인가? (각 3점 x 5개 = 15점)

```
#include <iostream.h>
class B{
public:
    B(int j=0): i(j){}
    virtual void print()
    {cout<< "B::" << "i=" << i << endl;}
    void print(char *s)
    {cout<< "B::" <<s << i << endl;}
private:
    int i;
};

class D: public B {
public:
    D(int j=0): B(5),i(j) { }
    void print()
    { cout << "A::" << "i=" << i << endl;}
    int print(char *s)
    {cout << "A::" << s << i << endl;
    return(i);}
private:
    int i;
};

main() {
    B b1, b2(10), *pb;
    D d1, d2(3), *pd = &d2;
    b1.print(); // (a)
    d2.print("d2.i="); // (b)
    pb=pd; pb->print(); // (c)
    pb->print("d2.i="); // (d)
    pd->print("d2.i="); // (e)
}
```

class의 member function의 body가 null인 함수(subclass 만들때 틀로 사용)

class의 용도 1. 객체만들기 2. 다른클래스 만들기
static message to method binding이기 때문에 필요

5. C++언어 (static message binding)의 abstract base class에서는 member function의 body
부분을 null (예를 들면, virtual foo() = 0)로 선언한다. (각 5점 x 2 = 10점)

- (a) C++ 언어에서 이런 선언을 하는 이유를 설명하라. (5점).
- (b) Dynamic message binding을 사용하는 Smalltalk 언어에서도 이런 기능이 필요한가?
그 이유와 함께 설명하라. (5점)

런타임때 찾아가니까 필요없음

6. 다음은 어떤 언어를 설계 시에 언어의 특성 및 구현 방법에 따라 5가지 언어로 분류한 것이다. 아래 물음 ((a) ~ (e))에 답하라. (21점)

Recursion 허용 여부	Scoping Rule	구현 방법	Activation Record에 저장되는 정보	
			공통적으로 저장되는 정보	추가 저장 정보
Recursion 허용	Static	Static chain	(a)	(b)-(1)
		Display		(b)-(2)
	Dynamic	Deep Access	(c)	(d)-(1)
		Shallow Access		(d)-(2)
Recursion 허용 안함	Static		(e) 함수 호출에 대한 Return address 저장 방법 (5점)	

fortran
activation record가 하나밖에 없음

7. Python & Prolog (40점)
다음과 같은 table 형식의 자료가 있다고 가정한다. 이 자료 구조를 저장하는 python 및 Prolog 코드는 다음과 같다.

birthday_table

이름 (name)	생일 (birthday)
홍길동	20110312
김복순	20170543
김서강	20001011
박수철	20010324

(a) 위의 자료 구조를 프로그램 내에 저장하기 위한 코드를 작성하라.
(가) Python의 dictionary를 이용하여 저장 (5점) birthday = {'홍길동' : xxx, }
(나) Prolog Fact를 이용한 저장 (5점) birthday (홍길동, xxx)

(b) (a)에서 정의한 table 정보에 대하여 사용자로 부터 이름을 입력 받아서 그 이름에 해당하는 생일을 출력하는 프로그램을
(가) Python으로 작성하라. (5점)
(나) Prolog를 질의 문을 작성하라.(10점

Enter name:홍길동
20110312

Python 수행 결과

Enter name 홍길동
Birthday = 20110312,
Name = 홍길동

Prolog 수행 결과

(c) (a)에서 정의한 table에 저장되어 있는 모든 데이터를 이름/생일 형식으로 출력하는 프로그램을
(가) Python for loop를 이용하여 작성하라. (5점)
(나) Prolog 질의 문을 작성하라. (10점)

홍길동 / 20110312
김복순 / 20170543
김서강 / 20001011
박수철 / 20010324

Python 수행 결과

홍길동/20110312
김복순/20170543
김서강/20001011
박수철/20010324
false

Prolog 수행 결과

<참고 1> Prolog 입출력 예제

writeln('서강대학교'), read(X), nl, writeln(X).

서강대학교

컴퓨터공학과

컴퓨터공학과

X = 컴퓨터공학과

?- writeln('서강대학교'), read(X), nl, writeln(X).

질의

<참고 2> Prolog primitive subgoal

- write('aa') : “aa”를 출력하는 primitive goal
- write(X) : X에 instantiation된 값을 출력하는 primitive goal
- read(X) : 입력을 받아서 그 값을 X에 instantiation (대치) 시키는 primitive goal.
- fail : 항상 fail하는 primitive subgoal

코드를 대체

1. 다음의 물음에 답하여라. (각 3점 x 7 = 21점)

- (a) Procedure와 Macro의 공통점은 무엇인가? 또한, 이들의 상대적인 장단점을 비교 설명하라.
- (b) Ada 언어는 parameter passing 방법으로 keyword parameter를 제공한다. 이 방법의 장단점을 설명하라. **순서를 바꿔도 되지만 실제 매개변수 이름을 다 알아야 함**
- (c) subprogram의 separate compilation 방식과 independent compilation 방식의 장단점을 설명하라. **sep - 함수들의 정의를 프로토타입에 따라 알아야 컴파일(다답제킹). 부분 컴파일. ind - 프로토타입 안해도 된 전체 다시 컴파일**
- (d) FORTRAN과 같이 subprogram의 변수도 statically allocate된다. 이런 언어 특성의 장단점을 설명하라. **no recursion**
- (e) Call-by-reference를 사용하는 언어에서 "call sub(a*b)"는 어떤 방식으로 컴파일되는지 설명하라.
- (f) static chain과 display는 궁극적으로 무엇을 구현하기 위한 방법인지 설명하고, C 언어는 이런 구현이 필요 없는 이유를 설명하라.
- (g) C++ 언어와 Java 언어에서 객체를 생성하는 방법과 생성된 객체가 저장되는 곳은 어디인지 각각 설명하라.

c - global 또는 stack
java - new로 생성 heap

2. 부 프로그램 (15점)

- (a) "계산"을 파라미터로 넘겨서 ascending 및 descending sort를 할 수 있는 bubble sort 서브프로그램 "HighOrder_Sort(OP, int Arr[], size)"를 c-like한 문법으로 작성하고, 이 서브프로그램을 ascending 및 descending sort하기 위하여 어떻게 호출하는지 호출하는 statement를 작성하라. (10점)
- (b) (a)에서 작성한 부프로그램이 generic sorting할 수 있게 하기 위하여서는 compile time 혹은 run-time에 어떤 일을 하여야 하는지 설명하라. (5점)

3. 다음과 같은 Pascal-like한 프로그램에서 아래와 같은 5가지 parameter-passing 방법들에 의하여 parameter가 전송된다고 가정하였을 때 다음 프로그램의 output은 무엇인가? 단, by result 및 by copy인 경우에 actual parameter의 주소는 값이 호출될 때 계산된다고 가정하고, 최종 값을 알 수 없는 경우는 "unknown"으로 답할 것. (3점 x 5 = 15점)

```
procedure BIGSUB
integer GLOBAL;
integer array LIST[1:2];
procedure SUB(PARAM);
integer PARAM;
begin
PARAM := 3;
GLOBAL := GLOBAL - 1;
PARAM := 5;
end
begin /* Main */
LIST[1] := 3;
LIST[2] := 1;
GLOBAL := 2;
SUB (LIST[GLOBAL]);
print(GLOBAL, LIST[1],
LIST[2]);
end
```

- (a) by value
(b) by result
(c) by copy
(d) by reference
(e) by name

4. Subprogram (3점 x 6 = 18점)

- (a) Overload Subprogram 기능을 제공하는 언어에서 함수 호출 (call)에 대한 함수 정의 (definition) binding은 무엇에 의하여 결정되는지 설명하라. (3점)
- (b) C++ 언어의 in-line 함수의 장점을 macro와 subprogram과 비교하여 설명하라. (3점)
- (c) 다음 각각 parameter passing 방법에서 actual parameter에 대한 주소 binding time이 언제인지 **구체적으로** 설명하라. (각 3점 x 3 = 9점)
- (가) call by copy (나) call by reference (다) call by name
- (e) Ada 언어는 function의 formal parameter는 항상 in-mode로 선언되어야 한다. 그 이유를 설명하라. (3점)

5. Python Function and Parameter Passing (각 3점 x 3 + 5점 = 14점)

```
def append_to_sequence (myseq):
myseq += (9,9,9)
return myseq

tuple1 = (1,2,3) # tuples are immutable
list1 = [1,2,3] # lists are mutable
tuple2 = append_to_sequence(tuple1)
list2 = append_to_sequence(list1)
print(tuple1, tuple2) /* (가) */
print(list1, list2) /* (나) */

def myfunc(n):
return lambda a : a + n
my1 = myfunc(2)
my2 = myfunc(3)
my3 = myfunc("lala")
print(my1(11), my2(11), my3("11")) /* (다) */
```

call by object

- (a) 위 프로그램을 출력 결과 ((가) ~ (다))를 써라 (각 3점 x 3 = 9점)
- (b) Python 언어는 Generic function 기능을 제공한다. 위의 함수 중에 어떤 함수가 Generic 함수인지 그 이유와 함께 설명하라. (5점)

6. 다음의 Python Coroutine 프로그램을 이용하여 아래 물음에 답하라. (10점)

```
① def countdown(n):
②     while n > 0:
③         yield n
④         n -= 1
⑤
⑥ for i in countdown(5):
⑦     print(i, end=" ")
```

- yield는 교재 coroutine의 return 값이 있는 resume과 같은 기능

- (a) 문장이 수행되는 순서를 써라. (예 ⑥ → ① → ② → ③) (5점)
- (b) 출력 값을 써라. (5점)

5 4 3 2 1

7. 다음의 C++ 프로그램과 Python 프로그램은 같은 기능을 하는 프로그램이다. (31점)

```
C++ 프로그램

#include <stdio.h>

class A {
protected:
    int i,j ;
public:
    A(int x, int y) {
        i = x; j = y;
    }
    virtual void printvar() {
        printf("%d %d\n", i, j);
    }
};

class B : public A {
protected: int i,j ;
public:
    B(int x, int y): A(11,22) {
        i = x; j = y ;
    }
    void printvar() {
        printf("%d %d\n", i, j);
    }
};

main() {

    A a(11,22), *pa ;
    B b(33,44), *pb;

    a.printvar();      /* (가) */
    b.printvar();      /* (나) */

    a = b ;
    a.printvar();      /* (다) */

    pa = &b;
    pa->printvar();    /* (라) */

}
```

```
Python 프로그램

class A :
    def __init__(self, i, j):
        self.i = i
        self.j = j

    def printvar(self):
        print(self.i, self.j)

class B (A):
    def __init__(self, i, j):
        A.__init__(self, 11,22)
        self.i = i
        self.j = j

    def printvar(self):
        print(self.i, self.j)

a = A(11, 22)
a.printvar()          /* (가) */

b = B(33, 44)
b.printvar()          /* (나) */

a = b
a.printvar()          /* (다) */

-----
# b = a
# b.printvar()
-----
```

- (a) C++ 프로그램에 의하여 프린트 되는 결과 ((가) ~ (라))를 써라 (각 3점 x 4 = 12점)
- (b) Python 프로그램에 의하여 프린트 되는 결과 ((가) ~ (다)) 써라 (각 3점 x 3= 9점)
- (c) Python 프로그램의 comment 처리된 부분 (점선으로 mark한 부분)의 comment를 제거하고 수행하면 어떤 결과가 나오는가? Error 없이 수행되면 프린트된 결과를 그 이유와 함께 설명하고, Error가 발생하면 어떤 이유로 error인지 설명하라. (10점)

8. 다음의 Prolog 프로그램을 각각 완성하라. (각 5점 x 4 = 20점)

```
(a) List reversing

append([], L, L).
append([X|L1], L2, [X|L3]) :- append(L1, L2, L3).

reverse([ ], [ ]).
reverse([H|T], L) :- reverse(T, L1), append(/* 여기를 채워라 */).

:- reverse([a,b,c,d],X).
X=[d,c,b,a]
```

```
(b) List Length

listlength([], 0).
listlength([_|T], L) :- /* 여기를 채워라 */.

:- listlength([a,b,c,d],X).
X=4
```

```
(c) Subset Test

member(X, [X|_]).
member(X, [_|Y]) :- member(X,Y).

subset([],_).
subset([X|L],Set) :- /* member()를 이용하여 여기를 채워라.*/

:- subset([1,2,3], [a,b,2,c,1,3,4]).
true
:- subset([1,2,3], [a,b,c,1,2,4]).
false
```

```
(d) List Permutation

perm(List, [H|Perm]) :- delete(/*여기를 채워라 */), perm(Rest,Perm).
perm([], []).

delete(X, [X|T], T).
delete(X, [H|T], [H|NT]) :-delete(X, T, NT).

:- perm([1,2,3,4], X).
X = [1, 2, 3, 4]
X = [1, 2, 4, 3] /* NEXT를 요청하면 다음 permutation이 나온다. */
X = [1, 3, 2, 4]
.....
```

1. 다음의 물음에 답하여라. (15점)

- (a) Call-by-reference를 사용하는 언어에서 "call sub(a*b)"는 어떤 방식으로 컴파일되는지 설명하라. (3점)
- (b) C++ 언어와 Java 언어에서 객체를 생성하는 방법과 생성된 객체가 저장되는 곳은 어디인지 각각 설명하라. (각 3점 x 2 = 6점)
- (c) C++나 Java와 같은 객체지향언어는 다른 종류의 언어들 보다 재사용(reuse)가 쉬운 장점이 있다. 이런 소프트웨어 재사용을 용이하게 하는 언어적인 특성은 어떤 것들이 있는가? 대표적인 특성 2개를 재사용이 가능한 이유와 함께 설명하라. (각 3점 x 2 = 6점)

(15점)

2. 다음과 같은 C++프로그램을 이용하여 아래의 물음에 답하라. (10점)

```
class One {
    public: virtual int value() { return 1;}
};
class Two: public One {
    public: int value() { return 2;}
};

main() {
    Two x ;
    One &a = x ;
    One *b ;
    One c;
    b = &x ;
    c = x ;
    printf("values %d %d %d\n",
        a.value(), b->value(), c.value()) ;
}
```

- (a) 이 프로그램의 프린트되는 결과는 무엇인가? (5점)
- (b) 만약 virtual 키워드를 없애면 프린트되는 결과는 무엇인가? (5점)

(10점)

3. Recursion 및 nested subprogram 선언을 허용하는 가상의 C-like 언어를 이용한 아래와 같은 함수 정의 (integer 변수를 입력으로 받아서 float 값을 return하는 함수)에 대하여 아래 물음에 답하라. 단, 32-bit CPU 및 word alignment를 하지 않는다고 가정한다.

```
float sub1(int x) {
    float temp ;
    temp = 3.0 * (float) x ;
    return (temp) ;
}
```

- (a) 이 가상의 언어에 대하여 static scoping rule을 사용하고, 이를 구현하기 위하여 display 구조를 사용하는 경우 위 sub1() 함수 수행을 위한 activation record에 저장하여야 하는 정보 및 크기 (단위 : byte)를 모두 써라 (10점)

(b) 만약 이 언어를 dynamic scoping rule을 적용하고 deep access 방법을 이용하여 구현하는 경우에 위 sub1() 함수 수행을 위한 activation record에 저장하여야 하는 정보가 (a)의 경우와 다른 점을 써라. 즉, (a)의 경우와 비교하여 어떤 정보를 빼고 어떤 정보 (내용 및 필요한 byte 수)를 추가하여야 하는지 써라. (10점)
(20점)

4. 다음 각각 parameter passing 방법에서 actual parameter에 대한 주소 binding time이 언제인지 구체적으로 설명하라. (각 3점 x 3 = 9점)

- (a) call by copy
- (b) call by reference
- (c) call by name

(9점)

5. 아래 같은 C++ 언어를 Linux상에서 컴파일한 결과가 다음과 같다. 다음의 물음에 답하라. (31점)

```
#include <stdio.h>
class A {
private :
    double i ; char c ; int j;
    static int k;
public :
    A() {j = 1; c='a'; i = j;}
    int sub1(int a, char b) {
        a=a+2; return(a);}
};

main() {
    int m=3 ;
    char c='A';
    A a ;
    m = a.sub1(m,c) ;
    printf("%d\n", sizeof(a));
}
```

```
.file "test.cc"
.version "01.01"
gcc2_compiled.:
.section .rodata
.LC0:
.string "%d\n"
.text
.align 4
.globl main
.type main,@function
main:
.LFB1:
pushl %ebp
.LCF0:
movl %esp, %ebp
.LCF1:
subl $40, %esp
.LCF2:
movl $3, -12(%ebp)
movb $65, -13(%ebp)
subl $12, %esp
leal -40(%ebp), %eax
pushl %eax
.LCF3:
call __1A
addl $16, %esp
movl %eax, %eax
subl $4, %esp
movsbl -13(%ebp), %eax
pushl %eax
pushl -12(%ebp)
leal -40(%ebp), %eax
pushl %eax
call sub1__1Aic
addl $16, %esp
movl %eax, %eax
movl %eax, -12(%ebp)
subl $8, %esp
pushl $16
pushl $.LC0
call printf
addl $16, %esp
movl $0, %eax
leave
ret
```

```
.LFE1:
.Lfe1:
.size main, .Lfe1-m
.section
.align 4
.weak __1A
.type __1A,@function
__1A:
.LFB2:
pushl %ebp
.LCF4:
movl %esp, %ebp
.LCF5:
movl 8(%ebp), %eax
movl $1, 12(%eax)
movb $97, 8(%eax)
fildl 12(%eax)
fstpl (%eax)
popl %ebp
.LFE2:
.Lfe2:
.size __1A, .Lfe2-__1A
.section
.align 4
.weak sub1__1Aic
.type sub1__1Aic,@function
sub1__1Aic:
.LFB3:
pushl %ebp
.LCF6:
movl %esp, %ebp
.LCF7:
movl 8(%ebp), %eax
movl 12(%ebp), %eax
movl 16(%ebp), %eax
movb %dl, %dl
addl $2, %eax
movl %eax, %eax
popl %ebp
ret
```

(a) 이 컴파일러는 word alignment를 하고 있나? 그 판단 근거를 class A의 constructor function A() body 부분에 대한 assembly code를 이용하여 설명하라.(5점)

(b) 이 프로그램에 의하여 프린트되는 결과는 무엇인가? 그 계산 수식과 함께 설명하라. (5점)

(c) 다음 각각의 변수에 대하여 그 변수가 선언된 함수 activation record상에서 저장된 offset (단위 : byte)을 그 근거와 함께 써라. (11 점)

- (가) main::c (3점)
- (나) A::a (3점)
- (다) A::a.c (5점)

(d) A 클래스 객체인 a는 main()의 local variable이기 때문에 main() 함수의 activation record안에 저장된다. 이 프로그램 수행 시 함수 호출은 main() -> A::A()의 순서로 일어난다. 또한, A::A() 함수가 호출되면 A 클래스 객체인 a의 instance 변수 값들을 초기화 한다. 이 경우 호출된 A::A() 함수에서는 main() 함수의 activation record에 저장되어 있는 a 객체의 instance 변수 (예를들면, A::i 등)를 어떤 방식으로 접근하여 (주소 계산을 하여) 초기화하는지 해당 assembly 코드를 이용하여 설명하라. (10점)
(31점)

6. 아래와 같은 Pascal-like한 프로그램에서 아래와 같은 5가지 parameter-passing 방법들에 의하여 P()가 호출된다고 가정하였을 때, 이 프로그램의 수행이 끝난뒤 v[3]과 v[4]에 저장된 내용은 각각 무엇인가? 단, by result 및 by copy인 경우에 actual parameter의 주소는 호출 전에 계산된다고 가정하고, 최종 값을 알 수 없는 경우는 Unknown으로 답할 것. (4점 x 5 = 20점)

```
begin
  integer k
  integer array v[10]
  procedure P(a)
    integer a
    begin
      a := a + 1
      v[k] := a + 1
      k := 3
      a := a + 2
    end
  end
  v[3] := 6
  v[4] := 8
  k := 4
  call P(v[k])
end
```

- (a) by value
- (b) by result
- (c) by copy
- (d) by reference
- (e) by name

(20점)

1. Recursion 및 nested subprogram 선언을 허용하는 가상의 C-like 언어를 이용한 아래와 같은 함수 정의 (integer 변수를 입력으로 받아서 float 값을 return하는 함수)에 대하여 아래 물음에 답하라. 단, 32-bit CPU 및 word alignment를 하지 않는다고 가정한다.

```
float sub1(int x) {
    float temp;
    temp = 3.0 * (float) x;
    return(temp);
}
```

(a) 이 가상의 언어에 대하여 static scoping rule을 사용하고, 이를 구현하기 위하여 display 구조를 사용하는 경우 위 sub1() 함수 수행을 위한 activation record에 저장하여야 하는 정보 및 크기 (단위 : byte)를 모두 써라 (10점)

(b) 만약 이 언어를 dynamic scoping rule을 적용하고 deep access 방법을 이용하여 구현하는 경우에 위 sub1() 함수 수행을 위한 activation record에 저장하여야 하는 정보가 (a)의 경우와 다른 점을 써라. 즉, (a)의 경우와 비교하여 어떤 정보를 빼고 어떤 정보 (내용 및 필요한 byte 수)를 추가하여야 하는지 써라. (10점) (20점)

2. 아래와 같은 C++ 언어 프로그램에 의하여 출력되는 결과를 써라. (10점)

```
class D {
public:
    D() { printf("D()\n"); }
    D(int i) { printf("D(int)\n"); }
    D(double d) { printf("D(double)\n"); }
};

class A : public D {
public:
    A() : D(7) { printf("A()\n"); }
};

class B : public D {
public:
    B() : D(3.14) { printf("B()\n"); }
};

class C : public B, public A {
public:
    C() : B(), A() { printf("C()\n"); }
};

main() {
    C c;
}
(10점)
```

3. 아래 같은 C++ 언어를 Linux상에서 컴파일한 결과가 다음과 같다. 다음의 물음에 답하라. (21점)

```
#include <stdio.h>
class A {
private:
    double i; char c; int j;
    static int k;
};
```

```
.file "test.cc"
.version "01.01"
gcc2_compiled.:
.LC0:
.string "%d\n"
```

```
.LFE1:
.Lfe1:
.size main, .Lfe1-m
.section __1A, .align 4, .weak
```

public : A() {j = 1; c='a'; i = j;} int sub1(int a, char b) { a=a+2; return(a);} }; main() { int m=3; char c='A'; A a; }	.text .align 4 .globl main main: .LFB1: pushl %ebp .LCF0: movl %esp, %ebp .LCF1: subl \$40, %esp	.type __1A,@function __1A: .LFB2: pushl %ebp .LCF4: movl %esp, %ebp .LCF5: movl 8(%ebp), %eax movl \$1, 12(%eax) movb \$97, 8(%eax) fildl 12(%eax)
---	---	--

4. 다음의 Pascal-like한 프로그램에서 아래와 같은 5가지 parameter-passing 방법들에 의하여 P()가 호출된다고 가정하였을 때, 이 프로그램의 수행이 끝난 뒤 a[1]와 a[2]에 저장된 내용은 각각 무엇인가?
단, by result 및 by copy인 경우에 actual parameter의 주소는 호출 전에 계산된다고 가정하고, 최종 값을 알 수 없는 경우는 "Unknown"으로 답할 것. (5점 x 4 = 20점)

```

program test;
var i : integer;
    a : array[1..2] of integer;

procedure p(x, y : integer);
begin
    x := x + 1;
    i := i + 1;
    y := y + 1;
end;
begin
    a[1] := 1;
    a[2] := 1;
    i = 1;
    p(a[i], a[i]);
end.

```

- (a) by value
(b) by result
(c) by copy
(d) by reference
(e) by name

(20점)

5. Jensen's Device (10점)

다음과 같은 기능을 수행하는 오른 쪽 함수의 body 부분을 C-like 언어 syntax를 이용하여 작성하여라.

호출	Return 값
SUM (A, I, 100)	100*A
SUM (A[I], I, 100)	$\sum_{j=0}^{100} A[j]$
SUM (A[I]*B[I], I, 100)	$\sum_{j=0}^{100} (A[j] * B[j])$

```

float SUM (void ADDER,
           int INDEX,
           int LENGTH)
{
    /* 여기를 채워라 */
}

```

(10점)

6. 다음 프로그램은 Dijkstra의 Guarded command를 이용한 loop structure이다. 이 loop의 수행이 끝나고 난 후 q1, q2, q3, q4에 저장되는 값들은 각각 무엇인가 ? (10 점)

```
q1 := 22; q2 := 11; q3 := 33; q4 := 0;
do  q1 > q2 -> temp := q1; q1 := q2 ; q2 := temp ;
[] q2 > q3 -> temp := q2; q2 := q3 ; q3 := temp ;
[] q3 < q4 -> temp := q3; q3 := q4 ; q4 := temp ;
od
(10점)
```

7. 다음과 같은 C++ 프로그램을 이용하여 아래의 질문에 답하라. (10점)

(a) 이 프로그램은 error가 있다. 어느 문장이 왜 error인지 설명하라. (5점)

(b) 위의 error 문장을 comment처리하고 난후에 이 프로그램에 의하여 print되는 결과를 써라. (5점)

```
class AA {
public:
    int i;
    AA(int x) {i = x; }
    int operator==(AA& x) {
        printf("CLASS AA => Wn");
        return i == x.i;
    }
};

class BB : public AA {
public:
    int j;
    BB(int x, int y) : AA(x) {j = y;}
    int operator==(BB& x) {
        printf("CLASS BB => Wn");
        return (i == x.i) && (j == x.j);
    }
};

main() {
    AA aa(3);
    BB bb(4,5);
    if (aa == bb) printf("1st TEST Wn");
    if (bb == aa) printf("2nd TEST Wn");
    if (bb == bb) printf("3rd TEST Wn");
}
```

(10점)

8. C++ 언어 (static message binding)의 abstract base class에서는 member function의 body 부분을 null (예를 들면, virtual foo() = 0)로 선언한다. Dynamic message binding을 사용하는 Smalltalk 언어에서도 이런 기능이 필요한가 ? 그 이유와 함께 설명하라. (10점)

(10점)