

2018년도 1학기 소프트웨어입문설계 기말고사 (2 / 2교시)

반: (월수반 / 화목반) 학번: _____ 이름: _____

- 2교시는 여러분이 우리 수업의 내용을 얼마나 잘 가지고 있는지 평가합니다.
각 문제 내용은 우리 수업 표준 환경을 가정하여 적혀 있으며 채점 기준 또한 수업자료에 기반합니다.
- 이번 시험에서는 여러분의 의견을 묻는 문항들이 존재합니다('의견'이라 표시해 둬). 여러분의 의견이 수업 시간에 논의했던 것(과거의 선배들이 결정한 것)과 일치한다면 정답입니다. 그렇지 않다 하더라도 문항의 관점(주로 '시점' 관련)을 벗어나지 않는 선에서 타당한 서술을 하는 경우 그에 해당하는 부분점수를 받을 수 있습니다. 관점 벗어남 여부 및 서술의 무게(득점량)는 출제자인 강사에 의해 측정되므로, 가급적 수업 내용 또는 '다른 문항에서 보이는 이야기들'과 연계하여 답안을 적는 것을 권장합니다.
- '정확하게'가 적혀 있는 문제는 답안을 한 글자도 다름 없이 정확하게 적어야 합니다.
단, 한국어/영어 단어 대신 (수업자료에 있던) 영어/한국어 단어를 적는 것은 가능합니다.
- '모두'가 적혀 있는 객관식 문제는 답이 1개 이상임을 의미합니다.
해당 문제에서 답을 고를 때마다 (배점 / 답 수)만큼 점수를 얻으며,
답이 아닌 것을 고를 때마다 (2 * 배점 / 답 수)점씩 감점됩니다(그렇다고 음수로 떨어지진 않아요).
- 2교시 시험은 50분 동안 진행되며 만점은 50점입니다.

1. 단어 **runtime**에 대한 설명으로 알맞지 않은 것을 모두 고르세요: (2점, 2분) ----- ()
 - 1) Python interpreter도 **runtime** (system)의 일종이라 말할 수 있습니다.
 - 2) C 컴파일러는 **runtime**에 **선언**을 읽고 적절한 숫자 값을 정합니다.
 - 3) 내가 작성한 main()은 C **runtime**에 의해 호출된다 말할 수 있습니다.
 - 4) 내 프로그램이 메모리 위에 올라 실제로 실행되는 시점은 **runtime**입니다.
2. 다음 중 C의 기본 형식이 아닌 것을 모두 고르세요: (2점, 2분) ----- ()
 - 1) int
 - 2) int *
 - 3) char [3]
 - 4) single (*)[3]
 - 5) double
3. C 컴파일러는 **상수** 개념을 적극적으로 활용하여 **컴파일**을 수행합니다. 예를 들어, 프로그래머가 적은 수식 $2 + 3$ 은 **compile** time에 수식 5로 간주될(미리 계산될) 수 있습니다. 이렇게 미리 해 두어도 괜찮은 이유를 30자 내외로 설명해 주세요: (4점, 5분)
()
4. 위 문항에 이어서, 변수 개념과 대비했을 때 C 컴파일러가 **상수** 수식을 **컴파일**하는 방법 및 그 장점을 단어 '명령어(instruction)'를 포함하여 30자 내외로 설명해 주세요: (총 4점, 8분)
()

5. (의견)사용자와 소통하는 프로그램의 경우 실행의 구성 요소들 중 반복에 해당하는 흐름을 갖추고 있어야 합니다. 이렇게 만들어 두어야 하는 이유에 대한 여러분의 의견을 서술해 주세요: (4점, 3분)

➤ Hint: 사용자(사람)은 컴퓨터(CPU)에 비해 매우 느려요.

()

6. 수업시간에 다룬 **정의**의 두 가지 의미와 이를 '누가' '언제' 수행하는지를 각각 적어 주세요: (총 3점, 3분)

의미	누가	언제

(7 ~ 9) 아래 두 명제를 보고 각 물음에 답해 주세요:

명제 1: Python에서는 이름 사전을 사용하여 '이름에 대한 위치 값'을 다루었고, C에서는 그런 거 없었다.
명제 2: Python에서는 메모리 위의 각 object(칸)에 형식 정보가 들어 있었고, C에서는 그런 거 없었다.

7. 명제 1과 6번 문항의 답안을 감안했을 때, 여러분이 수식으로서 적은 어떤 이름을 **컴파일**할 때 C 컴파일러가 반드시 고려해야 하는 것으로 가장 알맞은 것을 하나 고르세요: (2점, 1분) ----- ()

- 1) 여름방학 시작 날짜
- 2) 코드 짠 프로그래머가 얼마나 trustable한 지
- 3) 그 이름에 대한 선언
- 4) Python과의 라이벌 의식

8. 명제 1, 2와 7번 문항의 답안을 감안했을 때, 여러분이 수식으로서 적은 어떤 이름에 대해 C 컴파일러가 어떤 방식으로 '그 이름의 형식'을 **컴파일** 과정에 반영하는지 20자 내외로 설명해 주세요: (4점, 5분)

()

9. (의견, 어려움)명제 1, 2와 7, 8번 문항의 답안을 감안했을 때, Python과 C는 수식으로서 적은 이름을 다루는 방법이 완전히 다릅니다. 이러한 차이점으로 인해 두 프로그래밍 언어가 서로에게 갖는 장점으로 적합하다 생각하는 점을 간략히 서술해 주세요. (6점, 7분)

- hint: 예시로 구경해 본 상황들을 떠올려 보면 무언가 또렷한 가능/불가능성이 보일 거예요.
- hint: '수식으로서 적은 이름'을 중심에 두면 좋아요. 각 언어별 장점 한 가지씩만 서술해 주면 돼요.

()

10. C의 automatic 개념과 static 개념 사이의 차이점을 하나 짚고, automatic 변수와 static 변수에 대해 컴파일러가 **정의**를 수행하는 방법의 차이를 각각 20자 내외로 설명해 주세요: (총 5점, 5분)

개념상 차이: ()

방법상 차이: ()

11. 아래 코드에서, func()가 실질적으로 **배열** matrix의 각 칸을 다루고 있음에도 func()의 인수를 왜 '포인터' 형식으로 **선언**했는지, 왜 'int 세 칸짜리 **배열**에 대한' **포인터** 형식으로 **선언**했는지를 각각 20자 내외로 설명해 주세요. (총 7점, 4분)

```
void func(int (*m)[3])
{
    int row, column;

    for ( row = 0; row < 3; row++ )
        for ( column = 0; column < 3; column++ )
            m[row][column] = row + column + 1;
}

int main()
{
    int matrix[3][3];

    func(matrix);
}
```

인수를 '포인터' 형식으로 **선언**한 이유:
()

인수를 'int 세 칸짜리 **배열**에 대한' **포인터** 형식으로 **선언**한 이유:
()

12. '어떤 캐릭터의 능력치'와 같이 여러 값들을 한꺼번에 모아 두어야 의미를 갖는 상황에서, 우리는 **구조체**와 **배열**의 두 개념들중 하나를 선택적으로 사용하여 해당 목표를 달성할 수 있었습니다. 이 때 **구조체** 개념을 사용(**구조체 정의**를 통해 새로운 **구조체** 이름을 도입하여 사용)함으로써 가질 수 있는 상대적 장점 두 가지를 각각 단어 '멤버'와 'offset'을 포함하여 20자 내외로 설명해 주세요: (총 7점, 5분)

장점 1: ()

장점 2: ()