C Programming (W8)

Welcome!!

Please check attendance individually.

(Mobile App)

Things to do today

01 Notice: Attendance & Course evaluation

02 Lecture Notes (Ch.7)

03 Requirements:

Attendence

If you are 15 minutes late for class, the attendance system will automatically check for absence.

In the future, when requesting correction of an attendance system error, correction will not be made unless you prove your attendance time. (e.g., login time, time capture, etc.)

* Grades are determined based on relative evaluation.

Course evaluation

Previous

Course evaluation	Distribution of points	Note
Attendance	20 points	-1 point per absence
Codyssey	40 points	20 essential problems: 2 point per a problem
	Extra points	60 optional problems: 0.5 point per a problem
	20 points	Contribution (Peer evaluation & review)
Final exam	20 points	Open book
Total	**100**	**Complete Codyssey 80 problems: A+**

* Grades are determined based on relative evaluation.

Course evaluation

Changed!!!

AI & Big Data

Course evaluation	Distribution of points	Note
Attendance	20 points	-1 point per absence
Codyssey	30 points	10 essential problems: 3 point per a problem (Q.2 ~ 11)
	Extra points	9 optional problems: 0.5 point per a problem (Q.12 ~ 20)
	10 points	Contribution (Peer evaluation & review)
Practice & Quiz	20 points	Practice: Every class (2 ~ 5 practices) - send an email by the same day with screen capture of results Quiz: In class, depends on schedule
Final exam	20 points	Open book
Total	**100**	*** if complete Codyssey 80 problems: A+**

* Email including C-section, ID, Name

* Mac: Command (⌘) + Shift + 4
* Windows ⅁ + Shift + S

* Grades are determined based on relative evaluation.

The number of questions is the same, but the question items may change.

Subproject	SLearning Course	Problem		Essential
Step 1: Audition for Cody Enter	**Process 1:** Wasteland with Value, Magratea (Standard Input/Output)	01 Question 1 Introduce myself		X
		02 Question 2 Project Kick-Off!	1	O
		03 Question 3 Children who became Milliways candidates	2	O
		04 Question 4 Who will choose the one who will hold the key to destiny?	3	O
Step 2: 8-Step Training Program	**Process 2:** Sprouts Blooming in the Wasteland (Multidimensional Arrays)	01 Question 1 8-step training program	4	O
		02 Question 2 Dumbass, the problem is physical strength!	5	O
		03 Question 3 My Basic Workout Routine	6	O
	Process 3: Wounds Heal in the Gardener's Hands (Structures)	01 Question 1 A Fight with Yourself	7	O
		02 Question 2 Facing Trauma	8	O
		03 Question 3 There is No Way to Escape Your Own Ghosts	9	O
	Process 4: Temperature of Language (Pointers)	01Question 1 How Good is My Korean?	10	O
		02 Question 2 Consonants and Vowels		X
		03 Question 3 Basic Grammar Learned through Puzzles		X
	Process 5: Temperature of Sound (Function Pointers, Pointer Operations)	01 Question 1 Facing the Present		X
		02 Question 2 Finding My Voice		X
		03 Question 3 Those Who Realized the Principle		X
	Process 6: A Body Like Dry Firewood is Reborn (File Input/Output)	01 Question 1 Body and Mind Separately		X
		02 Question 2 Surrendering Your Body to the Pattern		X
	Process 7: I'm Not the Same as I Was Yesterday (Dynamic Memory Allocation)	01 Question 1 People Make People		X
	Process 9: The Distance I Can Reach Out (Preprocessing, External Variables, Split Compilation)	01 Question 1 Shadow Life		X
Step 3: Debut	**Process 10:**			10

Statement

In **C**, a **statement** is a complete instruction that tells the computer to do something.

It usually ends with a **semicolon (;)**. Compound Statement / Block. { }

Types of Statements in C:

1. **Declaration Statement:** Used to declare variables.

```
int a;
float pi = 3.14;
```

2. **Control Flow Statements:** Control the flow of execution.

- **Conditional statement:**

```
if (a > b) { ... }
  else {...}
```

- **Iteration Statement (Looping):**

```
for (int i = 0; i < 10; i++) { ... }
  while() {...}
```

Statement

C에서 문장은 컴퓨터에게 어떤 작업을 하라고 지시하는 완전한 명령어.

일반적으로 세미콜론(;)으로 끝남.

복합 문장/블록은 { } 사용.

C의 명령문 유형:

1. 선언문: 변수를 선언하는 데 사용.

```
int a;
float pi = 3.14;
```

2. 제어 흐름 문: 실행 흐름을 제어.

- 조건문

```
if (a > b) { … }
   else {…}
```

- 반복문

```
for (int I = 0; I < 10; i++) { … }
while() {…}
```

See you next week!
DO NOT miss the classes