

Object Oriented Programming and Practice, Midterm Exam

DCCS213(02)

Your ID#: []

Your Name: []

Read each question carefully and do the programming! Be sure to follow the instructions below!**[Instruction] 안내**

(1) **Javadoc comment** and **standard naming rule** are essential for all program (**If failure -1 point**).

(Javadoc 코멘트와 스탠다드 네이밍 룰은 이제 말하지 않아도 기본입니다. 실패 시 각 -1점 감점.)

(2) Please compress (.zip) and submit this report (.docx), and the entire Java code (.java file) you wrote into the BlackBoard.

(이 시험지와 답변에 사용된 자바 코드를 압축하여 블랙보드에 지정된 시간까지 반드시 제출하십시오.)

Midterm: 16:00 PM ~ 17:30 PM (90 Min.)

*I highly recommend that you start submitting to the Blackboard at 17:25PM, at least 5 minutes before the end. If you do not submit by the given time, you will receive 0 points.

(시험 끝나기 5분전인 5시 25분에는 블랙보드에 제출을 시작하길 권장합니다. 5시 30분에는 제출할 수 없도록 닫히며, 제출을 못했을 시 당연히 0점 처리됩니다.)

(3) The Java code should be executable when the TA runs. The submitted compressed file (.zip) must be named Midterm_YourID_Yourname.zip.

(자바 코드는 TA가 돌렸을 시 깔끔하게 돌아가야 하며, 제출될 압축파일은 반드시 Midterm_YourID_Yourname.zip 로 명명하여 제출하십시오.)

(4) There is a partial score. Even if you can't resolve it completely, I hope you can try it as far as you can.

(부분 점수가 있으므로, 완벽하게 해결하지 못하더라도 할 수 있는 만큼 시도해 보길 바랍니다.)

(5) There are **5 questions** in total, but I don't expect you to resolve all the questions within the given time. I hope you select the problem you can solve and solve it quickly. "Choice and focus" are also one of the programmer's core abilities. If you've followed the class normally and know your proficiency, I'm sure you'll do it well.

(총 5개의 질문이 있지만, 이것을 제한 시간내에 모두 해결할 수 없을 것입니다. 본인의 수준을 고려했을 때, 풀 수 있는 문제를 제한시간내에 가급적 많이 해결하십시오. "선택과 집중"은 프로그래머의 핵심 능력 중 하나입니다. 수업을 정상적으로 따라왔으며, 자신의 프로그래밍 숙달도를 알고 있는 학생이라면, 이를 잘 해낼 수 있을 것입니다.)

(6) To prevent abuse of ChatGPT, pattern recognition for issues is left to the developer role in most questions. Through this opportunity, I want you to directly feel in what areas you can have an edge over other developers in the AI era. "Raise an issue/Pattern Recognition/Requirement Segment" etc. ChatGPT is still far from perfect and can't beat a human developer. In other words, at this moment when everyone can use ChatGPT, the era has arrived where it is also very important who can efficiently divide and ask questions first. Fuse your abilities and technological advancements to show your best performance!

(ChatGPT의 남용을 방지하기 위해 대부분의 질문에서 문제에 대한 패턴 인식은 개발자 역할에 맡겨졌습니다. AI시대에서 다른 개발자에 비해 여러분이 어떤 부분에서 우위에 설 수 있는지를 이번 기회를 통해 직접적으로 느껴보기를 원합니다. "문제 제기 / 패턴 인식 / 요구사항 분할 능력" 등은 ChatGPT가 아직까지 인간 개발자를 뛰어 넘을 수 없습니다. 즉, 모두가 ChatGPT를 사용할 수 있는 이 순간에, 누가 먼저 효율적으로 문제를 분할하여 질문 할 수 있는지 여부도 매우 중요한 시대가 도래했습니다. 당신의 능력과 기술 발전을 융합하여 최고의 성과를 보여주십시오!)

(Q1) Please develop a decryption program that can decrypt the binary file given below based on the format described in the codebook. Binary files can be restored only by the order of the given codebook, and the following **two results should be obtained.** **The first is to get the same result as the screenshot** to check if the restore was successful. **Second, separate each different form, save it in a human readable file format, respectively, and submit these files (5 files).** **(25 Points).**

(아래와 같이 주어진 Binary file을 codebook의 형식을 바탕으로 해독할 수 있는 복호 프로그램을 개발하십시오. 바이너리 파일은 주어진 코드 북의 순서에 의해서만 복원될 수 있으며, 아래와 같은 두 가지 결과물을 얻어야 합니다. 첫번째는 스크린샷과 동일한 결과를 얻어서 복원이 성공했는지를 확인하십시오. 두번째는 각각의 서로 다른 양식을 각각 분리하여 사람이 읽을 수 있는 파일 형식으로 저장하고 파일들을 제출 하십시오 (총 5개의 파일).)

<Input>

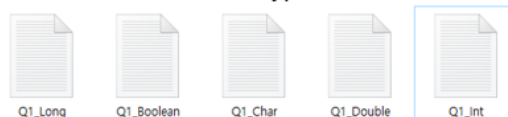
[Q1.data] Binary file

[Q1_CodeBook.data] codebook for cryptography

<Output>

```
The sum of all data of type double as a result of decryption is: 5986.477978998308
=====
# of integer values: 11939
# of boolean values: 11877
# of double values: 11955
# of char values: 11875
# of long values: 11836
```

[Outcome 1] To check if the decryption was successful, print the sum of all double values and the number of values of each type as above.



Q1_Int.txt	Q1_Boolean.txt	Q1_Double.txt	Q1_Char.txt	Q1_Long.txt
1 1832359680	1 true	1 0.9134358090941482	1 A	1 1859654906584037900
2 1586840465	2 false	2 0.19877939654072418	2 L	2 -3992804692506318686
3 706663334	3 true	3 0.7179282167952834	3 B	3 7004933895813745171
4 684101284	4 true	4 0.25739747025569526	4 E	4 -4326404132910526934
5 -1998994167	5 false	5 0.9939611449891282	5 L	5 6367311866054759106
6 886898449	6 false	6 0.26713231381869906	6 S	6 -6479422646462186477
7 2001087797	7 true	7 0.4333529573280621	7 O	7 8171451586954515451
8 832935172	8 false	8 0.7955520734121897	8 K	8 5726939552939717244
9 951746324	9 false	9 0.5448817328249995	9 X	9 5137037700605447497
10 1329552283	10 false	10 0.7928566249054845	10 V	10 -2884724901817857917
11 1158926529	11 false	11 0.2238408438556998	11 Y	11 1864018458841371482
12 1132781707	12 false	12 0.894488118167767	12 U	12 -7489028561914394345
13 1329338780	13 true	13 0.4996538906228991	13 O	13 199905723643018926
14 -1671157611	14 false	14 0.9411752436658591	14 U	14 6751728643103892337
15 1998092065	15 false	15 0.5319056508140191	15 A	15 8515959517570471803

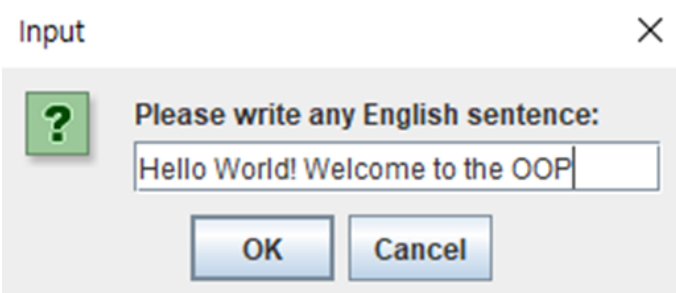
[Outcome 2] Write each type of value separately in human-readable text format to a separate files as above.

[Result Screenshot]

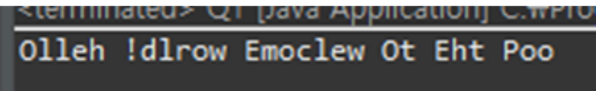
(Q2) Please write a program **Q2.java** that can receive English sentences and print them in reverse order word by word, as shown in the example below. In the example below, look for the **case rules** and be careful to get the same result. (15 Points).

(다음과 같이 영어 문장을 받아서 단어 별로 역순으로 출력할 수 있는 프로그램 Q2.java를 작성 하십시오. 아래 예제에서 대소문자 규칙을 찾고 동일한 결과를 얻도록 주의하십시오.)

<Input>



<Output>



[Result Screenshot]

(Q3) We want to create a used car management program. **Write Automobile & Q3.java main classes** that satisfies the following requirements **(25 Points)**:

(Kor. 중고차 관리 프로그램을 만들고자 합니다. 아래의 요구조건을 만족할 수 있도록 Automobile 클래스와 Q3 메인 클래스를 구현 하십시오.)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Model	Miles/(US	Number	Displacem	Gross hor	Rear axle	Weight (1	1/4 mile	Engine (0	Transmiss	Number c	Number c
2	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
3	Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
4	Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
5	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
6	Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
7	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
8	Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
9	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
10	Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
11	Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	1	0	4	4
12	Merc 280C	17.8	6	167.6	123	3.92	3.44	18.9	1	0	4	4
13	Merc 450SE	16.4	8	275.8	180	3.07	4.07	17.4	0	0	3	3
14	Merc 450SL	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
15	Merc 450SLC	15.2	8	275.8	180	3.07	3.78	18	0	0	3	3
16	Cadillac Fleetwood	10.4	8	472	205	2.93	5.25	17.98	0	0	3	4
17	Lincoln Continental	10.4	8	460	215	3	5.424	17.82	0	0	3	4
18	Chrysler Imperial	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4

(Step 1) Implement an Automobile class that can load information from a Q3.Data (tab-delimited) file and turn it into an individual object. All parts of a given material must be reflected in the object. (Kor. Q3.Data(탭으로 구분) 파일에서 정보를 불러와서 이를 개별 객체로 전환할 수 있는 Automobile 클래스를 구현하십시오. 주어진 자료의 모든 부분이 객체에 반영되어야 합니다.)

(Step 2) Implement a constructor that allows developers to create additional car objects from code. At this time, actively utilize at least 3 overloading constructors so that if a specific value is not known, it can be treated as a "NA" value. (Kor. 개발자가 코드에서부터 추가적으로 자동차 객체를 생성할 수 있는 생성자를 구현하십시오. 이 때, 특정한 값이 알려지지 않았다면 "NA" 값으로 처리될 수 있도록 최소 3개의 오버로딩 생성자를 적극적으로 활용하십시오.)

(Step 3) Randomly generate 100 new cars from the main class Q3. It's okay to give a completely random car model name (duplication is not allowed). Implement the genAutomobile() method so that all other feature values can be randomly generated within the existing data range in Q3.data file.

(Kor. 메인 클래스 Q3에서 100대의 새로운 자동차를 랜덤하게 생성하십시오. 자동차 모델명은 완전하게 랜덤하게 지어도 무방합니다 (중복은 불허합니다). Q3.data 파일의 기존 데이터 범위 내에서 다른 모든 특성 값이 생성될 수 있도록 genAutomobile() 함수를 구현하십시오.)

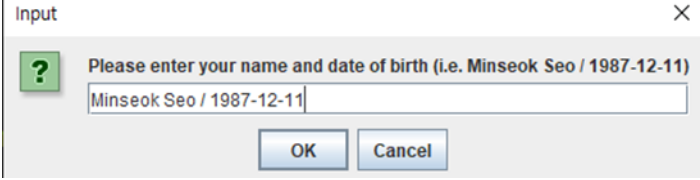
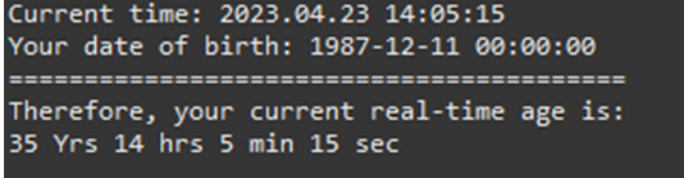
(Step 4) Finally, **design/implement a CUI that allows used car managers to easily search for various conditions of vehicles existing in the data.** For example, create an interface that receives a condition such as "*Miles > 20 and Number of cylinders == 4*" and outputs cars that meet the condition. **Since it is impossible to implement a perfect interface in time, all you have to do is describe the direction of the UI you want to ultimately develop and complete the development of the prototype to show that the vehicle is being searched for.** (Kor. 마지막으로, 중고차 관리자가 전산에 존재하는 차량들의 다양한 조건을 쉽게 검색할 수 있는 CUI를 디자인하십시오. 예를 들어, "*Miles > 20 and Number of cylinders == 4*" 등의 조건을 입력받아, 해당 조건의 자동차들을 출력해주는 기능을 만드십시오. 시간 상 완벽한 인터페이스 구현은 불가능하므로, 본인이 궁극적으로 개발하고자 하는 UI 방향성에 대해 기술하고 프로토타입까지 개발을 완료하여 차량이 검색되는 것을 보이면 됩니다.)

[In (Step 4), describe the most ideal CUI design considering the user:]

[A screenshot of the prototype of the CUI you designed.]

(Q4) Please develop a real-time age calculator (Q4.java) like the example below (**15 Points**).

(아래의 예제와 같은 실시간 나이 계산기를 개발하십시오).

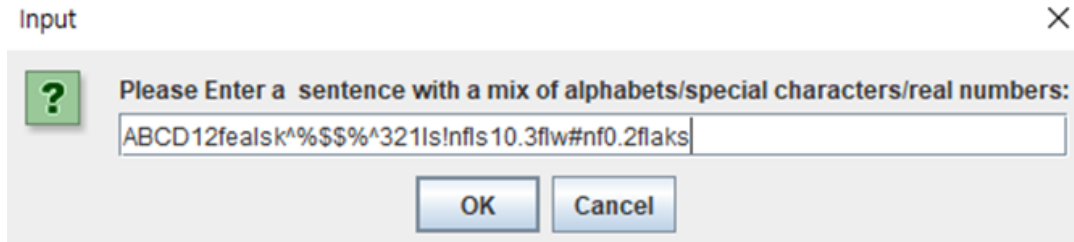
<Input>	<Output>
	

[Result Screenshot]

(Q5) As in the example below, develop a program Q5.java that can calculate all real numbers hidden in sentences. **(20 Points)**.

(아래의 예제와 같이 글자속에 숨겨져 있는 실수들을 모두 계산할 수 있는 프로그램 Q5.java를 개발하십시오.).

<Input>



Input

Please Enter a sentence with a mix of alphabets/special characters/real numbers:

ABCD12fealsk^%\$\$%^321ls!nfls10.3flw#nf0.2flaks

OK Cancel

<Output>

The result of adding all the real numbers in the input text is: $12 + 321 + 10.3 + 0.2 = 343.5$

[Result Screenshot]

-At the end- Well done for half of semester!