

[대학혁신지원사업]

[학습법]

2021-1학기  
원격수업 참여 사례 공모전
수상작 모음집

내가 참여한 수업이 최고!



아주대학교
AJOU UNIVERSITY

교수학습개발센터
Center for Teaching and Learning

차례

❖ I 등

이해랑(신소재공학과)

| 아트 콜라보레이션(Art Collaboration), 공학의 눈으로 나를 그리다1

❖ 2등

송용혁(전자공학과) | 대면인 듯 대면 아닌 대면 같은 비대면9

홍수빈(심리학과) | 인지행동치료입문 비대면 수업 참여 후기12

❖ 3등

김서란(산업공학과)

| 현대자동차, 그리고 UAM - <전략기술경영> 강의를 듣고19

문현진(사학과) | 한국근세사 비대면 수업 참여 공모24

신선아(e-비즈니스학과)

| 조직변화 수업으로 ‘나’를 변화시키는 경영의 미학28

최유빈(응용화학생명공학과) | 문석 교수님과 뽀모들35

하정민(국방디지털융합학과)

| 다른 어느 과목보다 열정을 기울이고 최선을 다했던 객체지향프로그래밍 및 실습수업41

❖ 입선

김기훈(전자공학과) | 비대면으로 소통하는 법142

김유미(금융공학과) | 나를 찾는 여행146

김지영(문화콘텐츠학과)

| 비대면 수업에 대한 불만, <콘텐츠 기획의 인문학>으로 사라지다149

박보현(환경안전공학과)

| 아는 만큼 보인다 (현실에서 이론이 적용되는 순간!)153

박서혜(산업공학과)

어찌할 수 없는 영역인 코로나 VS 어찌할 수 있는 영역의 ‘나’	158
박소연(의학과) 비대면으로도 가능한 양방향적 소통	162
이지수(미디어학과) 비전공자의 교양 도전기	165
조성호(산업공학과) 부족한 환경 안에서 최고의 학습 시스템 구현	168
조신애(경제학과) 주어진 환경을 받아들이고, 극복하기까지	171

1등

아트 콜라보레이션(Art Collaboration), 공학의 눈으로 나를 그리다

교과목명 : 한국미술사

교수명 : 안지연

학과 : 신소재공학과

이름 : 이해량

1. 수업 참여 후기

3학년 1학기부터 시작된 코로나의 여파로, 비대면 수업에 임하는 것은 이번이 세 번째 학기였습니다. 4학년 1학기에 들어서며, 앞으로의 진로에 대한 고민도 많아지고, 뭔가 해야 한다는 초조함과 긴장이 늘 뒤따르게 되었습니다. 3학년 1학기, 2학기에는 급작스러운 상황에 적응하고, 그 당시 할 수 있는 최선을 다하는 것으로도 충분하다 생각했습니다. 하지만 4학년 1학기에는 비대면 상황에서도 뭔가 해나가야 한다는 압박이 자연스레 생겼던 것 같습니다. 온라인 학습의 장기화, 그리고 사회인에 가까워지는 상황 속에서, 제 자신에 대한 불안은 커졌고 스스로에 대한 모호함은 심화되는 상황이었습니다. 돌이켜보면, 현재 제가 마주하고 있는 상황을 차분히 바라보고, 저를 설계할 수 있는 시간, 그런 유연함이 가장 필요한 순간이 아니었나 싶습니다. 비대면 상황이 길어짐에 따라, 비단 저뿐 아니라 다른 학우들에게도 이는 상당히 중요하다는 생각이 듭니다. 방향을 다잡고, 잠시 숨을 고르며 앞으로 나아갈 수 있는 유연함, 이를 갖출 수 있었던 건 안지연 교수님의 <한국미술사> 수업을 수강하면서부터였습니다.

비대면 학습이 보편화된 후, 교수님들의 수업 방식은 크게 세 가지 양식을 따르고 있습니다. 실시간, 동영상, 실시간과 동영상의 혼용이며, 이 수업은 실시간으로 수업하되, 수업 이후 녹화 동영상을 강의노트에 게시해주는 방식이었습니다. 실시간으로 학습하면서 동영상 재시청이 가능하다는 점에서 학습의 편리함이 있었습니다. 또, 아침 9시 수업임에도 매일 실시간으로 진행되었고, 사전에 늘 학습 자료를 최신화하는 등 최선을 다해 수업을 준비하시는 것을 느낄 수 있었습니다. 학생들이 생각하고, 고민할 거리가 있는 수업을 만드시기 위해 매일 수업 직후에 과제물을 제

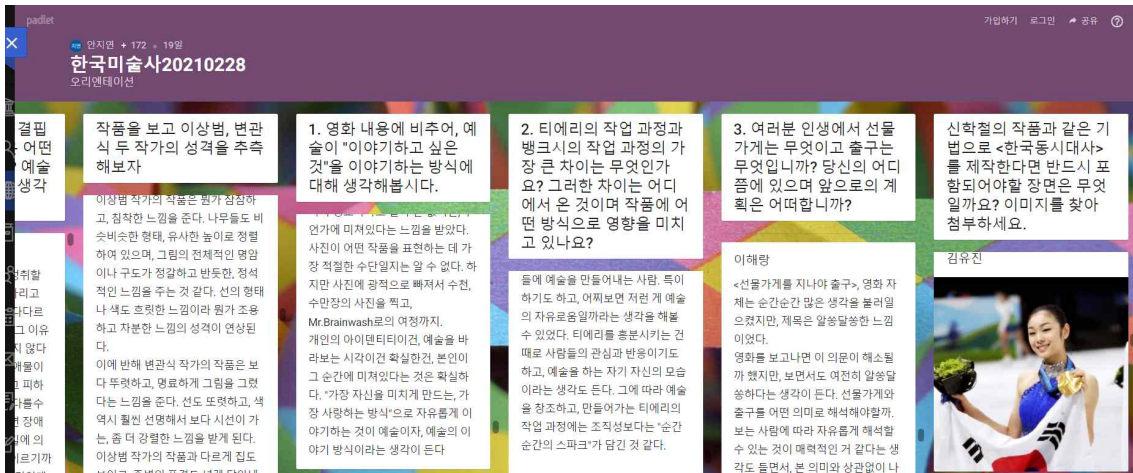
시해준다는 점도 특징적이었습니다. 또, 수업 중에는 학생들이 충분히 의견을 이야기할 수 있도록 채팅 및 마이크 사용을 자유롭게 허가하셨으며, 비대면 상황 속에서도 살아있는 수업을 받고 있다는 느낌을 받을 수 있었습니다.

3학기 동안 온라인 수업을 수강하며, 항상 긴장 상태에 놓여있었던 것과 달리, 이 수업에서는 오랫동안 학습의 여운이 남는다는 느낌을 받았습니다. 그 이유는 흘러가는 한국미술 수업이 아닌, 충분히 곱씹어볼 수 있는 적절한 과제물의 제출에 있습니다. 교수님께서서는 매번 수업이 끝난 후, 혹은 수업 중에 패들렛을 이용한 질문을 부여하셨습니다. 그날그날 수업과 관련된, 그 개념을 확장할 수 있는 질문들이었으며, 수업 전반을 되살려보고, 저만의 답을 제시할 수 있었습니다. 또한, 패들렛의 특성에 기반하여 다른 학우들의 생각과 이야기도 함께 공유할 수 있었습니다. 실시간으로 업데이트되는 답변들을 보며 때로 위안받기도 하고, 다양한 시각이 존재한다는 것에 신기함도 느낄 수 있었습니다. 블랙보드 시스템을 통해 파일을 제출하는 경우도 있었지만, 주된 과제는 패들렛을 통해 작성하였으며 수업 전과 직후, 매 수업 전까지 교수님을 포함하여, 수강생 모두와 교류하고 있다는 느낌을 받을 수 있었습니다. 비대면 상황 속에서 대면이 가지는 장점을 잘 살린 시스템을 통해, 더 열심히, 적극적으로 수업에 임할 수 있었습니다.

패들렛의 질문에 매번 답변을 달며, 저 자신을 생각할 수 있는 시간을 가질 수 있었습니다. 신소재공학과, 공대에 재학하면서 나는 잘하고 있는 걸까? 어디쯤에 와있을까, 라는 고민들을 해야 함에도 하지 못했던 순간도 있고, 휙쓸리듯 지나온 시간들이 분명 존재한다고 생각합니다. 이 수업은 그런 시간들을 하나로 단단히 묶어준, 쉽표이자 시작점이었다고 생각합니다. 수업을 위해 교수님이 준비하시는 영상자료 및 철저한 사전 준비를 바탕으로, 제 방향성을 고민할 수 있는 시간이 충분히 보장되었고, 순수하게 즐거운 마음으로 과제에 임할 수 있었습니다. 한 수업에 하나씩, 수업마다 부여되는 과제물들을 작성하며 그 시간만큼은 온전히 긴장을 내려놓을 수 있었습니다. 교수님이 제시하신 질문 중, 가장 인상적이었던 질문은 다음을 꼽을 수 있습니다. “당신의 선물가게는 어디이며 출구는 어디쯤이라고 생각합니까?”, 4년 동안 학교를 다니며, 제가 찾아야 하는 방향성에 대한 질문을 가장 명료하게 제시해준 순간이었습니다. 여기서 첫째, 이 수업의 특징으로 ‘즐거운 자기주도학습’을 이야기하고 싶습니다. 과제의 틀에 얽매이지 않고 상호 연결성을 가지되, 저에게 집중할 수 있는 시간, 그를 위한 학습 시스템과 수업 방식이 효과적으로 구비되어 있음을 느꼈습니다.

어떤 결과가 있다면, 뒤따르는 피드백 역시 필수적이라고 생각합니다. 인풋은 있

되 아웃풋은 없다면, 그 역시 올바른 과정이라 할 수 없으며, 100문제를 풀어도 채점과 풀이를 거치지 않는다면 무의미하다고 생각합니다. 제가 하고 있는 것에 대한 긍정/부정, 보다 발전할 수 있는 힘은 결국, 성실한 피드백에 있다고 생각합니다. 그런 관점에서, 이 수업의 두 번째 특징인 ‘즉각적인 응답성’을 얘기해볼 수 있습니다. 교수님께서서는 매번 학생들의 과제물을 소개하는 시간을 가지셨습니다. 그 덕분에 제 답변을 한 번 더 고민해보게 되었고, 학기 마지막까지 지치지 않고 달려갈 수 있었습니다. 우수한 답변으로 소개되면 뿌듯함을 느끼기도 했고, 기타 과제물에 대해서도 교수님이 의견을 남겨주시는 등 블랙보드의 활용 측면에서도 우수한 수업을 제공받을 수 있었습니다.



▲ 패들렛으로 제시된 질문 및 답변

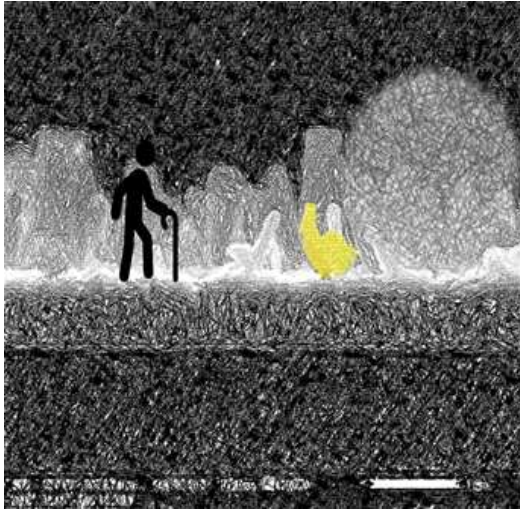
아트 콜라보 수업, 공학의 가치를 실현하다. 이 수업에서 제가 가장 강조하고 싶은 세 번째 특징은, ‘콜라보레이션’입니다. 한국미술은 예술의 영역이고, 공학은 기술의 영역이라, 어찌 보면 전혀 상상이 안 가는 조합이라 할 수 있습니다. 하지만 교수님께서서는 공학이 한국미술과 합쳐지는 사례를 실제로 수업시간에 소개해주셨고 (예: 추사 김정희_세한도 VR), 이외에도 한국미술이란 어느 하나로 국한되는 것이 아니라, 여러 영역과의 만남을 도모할 수 있다는 것을 강조하셨습니다. 기존에는 전혀 생각해보지 못했던 영역으로의 사고 확장을 가능하게 하셨으며, 실제 중간/기말 대체과제로도 개인의 전공과, 한국미술의 접점을 찾아 프로젝트를 수행하는 것을 부여하셨습니다. 다소 막막하고, 생소할 수 있는 주제임에도 제 전공 분야에서 예술 영역과의 만남을 도모할 수 있다는 게 상당히 흥미롭게 다가왔습니다. 또, 이는 전공 지식과 경험을 합칠 수 있는 방향을 고민하는 계기가 되었습니다.

길어지는 비대면 속에 다른 친구들을 비롯하여, 저 역시 많이 지쳐있었다는 생각

이 듭니다. 제 전공을 택하고, 여러 수업을 들어오면서 어느 순간 기계적으로 수업에 임하기도 했습니다. 하지만 <한국미술사> 수업에서는, 단순 문제풀이식이 아닌 대체과제로 평가를 진행하셨고, 교수님이 말하고자 하시는 바를 온전히 느낄 수 있었습니다. 교수님이 부여하신 중간/기말 대체과제를 통해 저는 신소재공학과 한국미술이 가지는 공통의 가치를 발견하는 한편, ‘나노 아트’ 라는 결과물을 기획하고, 실현할 수 있었습니다. 5월 즈음부터 매주 주말, 실험실에 가서 직접 나노 아트를 위한 나노 물질 합성 및 시편 촬영을 진행하였습니다. 그 과정에서 제 전공이 이렇게도 확장될 수 있다는 것에 신기함을 느끼는 한편, 저만의 아이디어를 제시해볼 수 있었습니다. 이는 전공과목의 설계실험이나 기타 이론 수업에서는 절대 해볼 수 없는 일이었으며, <한국미술사>를 계기로 제가 그리게 된, 저만의 결과물이라는 점에서도 굉장히 유의미한 경험이었습니다.

프로젝트를 진행해나가며 순간순간 생기는 의문들과 기타 질문사항에 대해 곧바로 답변을 남겨주신 교수님께 정말 감사드리고, 사소한 이메일에도 늘 귀 기울여 주신 덕에 하나의 수업을 함께 만들어간다는 것이 무엇인지 알 수 있는 한 학기였습니다. 수업의 마지막 즈음에는, 예술계에 종사하시는 예술인을 직접 초빙하여 줌으로 이야기를 듣고, 궁금한 것들을 질문할 수 있는 시간도 마련되었습니다. 몇 개월간의 수업 속에 흐트러질 수 있는 집중도를 제고하는 시간이었으며, 다수의 학생들이 남기는 양질의 질문을 통해 저 역시 질적인 학습을 도모할 수 있었습니다.

궁극적으로, 수업을 수강한 다수의 학생들에게 아트 콜라보레이션을 실천할 수 있는 기회를 제공하였으며, 이 경험을 통해 앞으로 보다 넓은 세상, 다양한 진로를 고민해보게 될 것 같습니다. 한 학기 동안의 <한국미술사> 수업은, 제 전공과 아트를 연결 짓고, 저를 고민해보게 만들고, 마침내는 저만의 생각과 결과물을 이끌어 낸, ‘아트 콜라보레이션(Art Collaboration) 수업’ 이라고 정의할 수 있을 것 같습니다.



▲ 직접 제작한 나노아트 결과물

하였다. 나노아트를 위한 전문적인 장비가 있는 것은 아니었기에 단순한 아이디어로 시작하였으
며, 우선, 초기의 기본적인 작동 제작 과정은 다음과 같다.²⁾

1. 테이프에 작은 크기로 (나노 장비로 볼 수 없게끔), 원하는 형상을 파낸다³⁾
2. 기판에 형상을 파낸 테이프를 붙인다.⁴⁾
3. 우리가 원하는 영역(테이프의 파낸 영역)을 제외하고, 나노물질 합성과 성장이 이뤄진다.⁵⁾
4. 육안으로 물질합성을 확인할 수 있으며, 이를 SEM 장비를 통해 이미지로 촬영하여 나노 단위에서 관찰할 수 있다.⁶⁾

간단하게 제시하자면 위와 같으며, 사용 장비는 아래와 같이 사진으로 덧붙여두었다.⁷⁾



몇주간의 나노아트 시도과정을 통해 얻은 결과물이다. (결과 제작을 위한 실험 과정의 상세한 설명은 제외하였으며, 오리 모양의 토기, 새를 본뜬 솥대의 형상, 그 밖에 연습용이 시도하였던 '아주' 글자 형상, 부처님의 모습 사진이다. / 이는 나노물질이 합성되던 하였으나 나노 장비를 통해 보기에 큰 사이즈였다. 물론 기판의 크기가 좌우 1~1.5cm 남짓이었기에 가능한 한 작게 파기 위해 노력을 기울인 결과물이다.)⁸⁾

▲ 나노아트 제작 과정 중 일부
(보고서 내용 발췌)

2. 나만의 학습 노하우

1. 나만의 보상 만들기 (단, 보상지점을 세분화하자)

좋은 성적을 냈을 때 부모님께 보상을 받았던 경험을 해본 적이 있을 것입니다. 저 역시 그랬던 기억이 있고, 그 보상을 위해 열심히 했던 적도 있습니다. 물론 이를 위해서는 적당한 원칙이 분명히 전제되어야 하지만, 많은 친구들이 스스로에게 지나치게 인색한 경우가 많습니다. 단위를 잘게 쪼개고, 목표 지점까지의 길을 세분화할수록 목적지가 명료해지고, 그만큼 원동력을 얻게 됩니다. 즉, 적절한 보상은 학습 효율을 증가시킬 뿐 아니라, 능동적인 학습에도 크게 기여하는 효과가 있습니다. 이는 특히 자기 관리가 힘든 비대면 상황에서 더욱 알맞은 학습 방법이라 할 수 있습니다. 이를테면 1시간 반 정도의 긴 동영상 수업을 수강할 때, 20분 단위 혹은 수업 흐름에 따라 파트별로 보상을 설정하는 것입니다. 동영상 수강이 밀리는 것으로 고민하거나, 수강 중에 동영상을 일시 정지하면서 자책감을 느끼는 경우가 종종 있습니다. 물론, 저 역시 그랬던 경험이 있습니다. 하지만 동영상을 20분 정도 집중해서 수강하고, 제가 보고 싶었던 동영상을 하나 시청하거나, 듣고 싶은 음악을 듣는 등 보상을 설정함으로써 이런 문제를 충분히 타파할 수 있었습니다. 또한, 이는 실시간 강의에서도 적용이 가능한 방법입니다. <한국미술사>의 경우, 아침 9시 수업으로 인해 늘 일찍 일어나야 한다는 어려움이 있었습니다. 하지만 일찍 일어나서 수업을 수강하고, 맛있는 아침을 먹는 등 작은 보상을 설정하는 것으로 꾸준한 학습을 도모할 수 있었습니다. 여기서 핵심은 보상 지점을 명료화하고, 세분화하여 설정

하는 것입니다. 너무 큰 목표를 세우고 보상을 설정할 경우, 빨리 지치는 문제가 발생할 수 있습니다. 보상에 명료한 기준을 정하는 것, 자신을 위한 당근을 알맞을 때 제공한다면, 종강까지 꾸준한 학습을 해나갈 수 있을 것입니다.

2. 나를 각인시키기 (우선, 작은 것부터 시작하자)

사람은 누구나 자신의 이름을 기억해주고, 알아줄 때 더 최선을 다하게 되고, 한 번 더 살펴보고 돌아보게 됩니다. 실제로 수업 중에 교수님이 언급을 하거나, 질문을 하는 등 저를 지정할 때 아무래도 더 몰입하는 효과가 생깁니다. 이를 활용하여, 수업시간에 저를 각인시키는 것은 굉장히 큰 학습 태도 변화를 불러일으킵니다. 예를 들어, A라는 수업에서 교수님이 저를 기억하고 제가 제출한 과제를 얘기하신다면, 자연스럽게 그 수업을 기다리게 되고, 스스로 만족할만한 결과를 내기 위해 노력하게 될 것입니다. 반면 B라는 수업에서는 그저 출석만 하고, 그 어떤 능동적인 참여를 하지 않는다면, 아무래도 수업시간의 집중도가 떨어질 것입니다. 중고등학생 시절, 앞줄, 특히 교탁 맨 앞자리의 친구들이 선생님과 많은 상호작용을 하고, 좋은 결과를 냈던 것을 떠올려보면, 이 효과는 더욱 확실해집니다. 나를 기억할 수 있게끔 적극적으로 참여하고, 그로 인한 학습 태도의 변화를 일으키는 것, 이는 결국 최선을 유도하고, 최고의 결과를 이끄는 가장 효과적인 방법이라 할 수 있습니다. 물론 참여하는 것이 익숙하지 않다면, 어렵거나 부담을 가질 수도 있을 것입니다. 하지만 저의 경우, 작은 것에서부터 시작하였습니다. <한국미술사> 수업을 수강하며, 출석을 부를 때, 과제를 제출할 때 혹은 수업이 끝난 후, 항상 인사를 열심히 하였고, 매 순간 성실하게 임하였습니다. 수업을 해주시는 교수님들께 감사 인사를 남기는 일 역시 별일 아닌 듯해도 큰 각인효과를 가집니다. 먼저 인사를 건네고 대답에 두려움을 없애는 것부터 시작한다면, 스스로 변화된 모습을 확인할 수 있을 것입니다. 저 역시 이를 통해 한 학기 수업을 설레는 마음으로, 끝까지 잘 수강해나갈 수 있었던 것 같습니다.

3. 수업 직후를 활용하기 (생각하는 것을 습관화하자)

사실 처음 비대면을 시작할 때는 저 역시 시행착오를 많이 겪었습니다. 헤이해지는 문제도 발생하였고, 학습 방향에 대해서도 갈피를 잡지 못했습니다. 하지만 차츰 안정을 찾고, 특히 이번 학기에는 스스로 방향성을 찾아갈 수 있었습니다. 이에 가장 큰 기여를 한 것은, 수업 직후 시간의 활용이라 할 수 있습니다. 많은 친구들이 공감하겠지만, 수강 과목 수가 많은 경우 교양과목에 몰두하기 어려운 상황에 놓이곤 합니다. 전공과목이 다양해지는 고학년일수록 그런 어려움을 겪게 되고, 이때 가

장 중요한 것이 수업 직후를 활용하는 것이라 할 수 있습니다. 저는 <한국미술사> 수업을 수강하며, 해당 날짜의 과제물은 되도록 수업 직후 고민해보고 정리하고자 하였습니다. 꼭 완벽한 제출물을 완성하지 않더라도 가이드라인을 잡아놓는다거나 필요한 부분을 정리하는 것만으로도 복습 및 학습 효율성을 제고하는 효과가 있습니다. 이는 다른 교과목에도 물론 적용될 수 있으며, 수업의 내용들이 조각조각 흩어지게 되는 것을 방지해줍니다. 단, 몰랐던 부분을 다시 보거나 그날의 내용을 정리하는 시간은 5 ~ 10분 정도면 충분합니다. 그 시간이 길어질 경우, 부담으로 다가올 수 있고, 장기적으로 유지하기에는 어려움이 뒤따릅니다. 사소한 듯해도 이 과정을 반복하다 보면 어느 순간 사고하는 것이 습관화되고, 누적 복습을 할 수 있게 됩니다. 집에서 이동 시간이 없는 만큼, 5분의 투자부터 시작해볼 것을 권하고 싶습니다.

4. 포스트잇 활용하기 (단, 눈에 보이는 곳에 붙여두자)

이는 고등학생 때부터 즐겨 활용하던 방식으로, 단순 노트 정리보다 효과적인 학습법이라 할 수 있습니다. 노트 정리를 하다 보면 어느 순간, 정리에만 몰두하게 되고, 그것으로 공부를 끝내는 경우가 많아집니다. 이는 오히려 학습의 질적인 면에서 부실할 수 있습니다. 튼튼한 벽돌집을 만들기 위해 벽돌 하나하나를 잘 쌓아야 하는 것도 맞지만, 점검을 하는 과정도 반드시 필요합니다. 그렇기에 제가 알고 있는 것들, 혹은 외워야 하는 내용들, 잘 와닿지 않는 부분을 작은 포스트잇에 적어두고, 눈에 잘 띄는 곳에 두는 것은 아주 효과적인 반복 효과를 가지게 됩니다. 외우기 위해 노력하기보다는 시각이 익숙해지게 만드는 것, 이는 교양과 전공 모두에 적용될 수 있으며, 일상에서 많은 내용이 떠오르는 것을 경험하게 될 것입니다.

5. 형광펜, 필기도구 활용하기 (다양한 색을 사용해보자)

공부는 눈에 바로 보이는 결과물이 없습니다. 시험을 치기 전까지는 내가 어디쯤에 있는지 파악하기 어렵고, 시험 직전에는 오히려 능률이 떨어지기도 합니다. 이 방법은 공부가 지겹거나 힘들 때 주로 사용했던 것으로, 아주대의 다른 친구들 역시 즐겁게 공부할 수 있는 방법이 될 수 있지 않을까 싶습니다. 공부를 할 때, 형광펜이나 필기도구를 다양한 색으로 준비하고 자주 바꿔보는 것입니다. 각 과목의 프린트를 보다 보면 집중이 흐트러질 때가 있고, 특히 교양과목은 프린트의 양도 방대한 경우가 많습니다. 그럴 때, 다양한 색상의 필기도구를 활용하면 한 가지 색을 사용할 때보다 재밌게 공부할 수 있습니다. 잠깐이지만 그 순간을 환기하는 효과도 있고, 그 자리에 조금이라도 더 오래, 성실하게 임하게 되기도 합니다. 꼭 중요도에

따라 색을 달리할 필요는 없으며, 조금이라도 흥미를 유발하는 것이 핵심적이라 할 수 있습니다. 저는 위 방법들을 통해 알찬 한 학기를 보낼 수 있었으며, 향후 수업에도 보다 유연하게 대처할 수 있을 것으로 생각합니다.

공부에는 왕도가 없습니다. 그렇기에 각자의 방법으로, 한 학기 동안의 제 학습 노하우를 참고하기도 하며, 현명하고 보람찬 학습을 해나가시기를 바랍니다. 즐겁고 행복하게, 무엇보다 건강하게 공부해나가시기를 바라며, 아주대 학생들 모두를 진심으로 응원합니다.

2등

대면인 듯 대면 아닌 대면 같은 비대면

교과목명 : 영어2

교수명 : 김현진

학과 : 전자공학과

이름 : 송용혁

1. 수업 참여 후기

대학교에 입학하고 처음 접해보는 비대면 원격수업은 기대 반, 우려 반이었다. 통학시간을 아낄 수 있어 더 많은 자습시간을 확보할 수 있고, 이해가 되지 않거나 놓친 부분은 다시 동영상 돌려 볼 수 있다는 점에서 비대면 수업은 장점이 있었지만, 교수와 학습자 간, 또는 학생들 간의 소통이 미흡하다는 점에서 학습 집중도가 떨어져 효율적인 학습이 이루어질 수 있을지 걱정이 되었다. 하지만 한 학기 동안 진행된 이 수업은 나의 우려를 불식시키기에 충분하였다. 이 수업에서 교수님은 우선 학생들 간 소통할 수 있는 기회를 많이 만들어주셨다. 원격수업이 진행되는 zoom 회의실에서 다시 소회의실을 따로 만들어 3 ~ 4명의 학생들로 조원을 만들어 주셨고, 이렇게 만들어진 조원들끼리 교수님이 내주신 질문에 대한 답변을 찾기 위해 서로 토의해 볼 수 있었다. 이를 통해 조원들과 함께 답을 고민해보고 자신의 생각을 상대방에게 전달하는 의사소통 과정을 통해 논리력을 키울 수 있었으며, 상대방의 말을 경청해야 하기에 학습 집중도가 높아져 자연스럽게 학습 효율 또한 증대시킬 수 있었다. 이러한 조원 토의 활동은 매주 진행되었으며, 형성되는 조원들도 매주 새롭게 바뀌었기 때문에 다양한 사람들과 소통해보며 다양한 관점으로 문제 상황을 바라볼 수 있었다. 원격수업에서 학생들과의 조원 활동이 가능할 것이라고는 예상하지 못했고, 다른 수업에서도 경험해보지 못했기 때문에 이 강의의 수업 방식은 정말 인상 깊었다.

하지만 상대방과 소통할 수 있는 기회는 여기에 그치지 않았다. 교수님은 아주대학교 Bb에 있는 토론학습 게시판을 이용하여 학생들이 서로 많은 의견을 나눌 수 있도록 도와주셨다. 그 예로 학기 초 간단한 자기소개를 이 게시판에 올렸는데, 많은 학생들이 댓글을 달아주어 비대면 환경 속에서도 서로 인사를 나눌 수 있었다. 또한 교수님께서 이 게시판을 통해 개인 발표 과제를 제출하게 하신 적이 있었는데

데, 이를 통해 다른 학생들이 제출한 과제를 보며 댓글을 통해 어떤 점이 좋았고 부족한 점은 무엇이었는지 알려줄 수 있는 피드백 환경이 조성되었다. 이처럼 학생 간의 소통 중심 수업을 통해 교수님이 가르치시는 내용을 외우기만 하는 피동적인 주입식 교육이 아니라 학생들끼리 고민해보고 협업하여 능동적으로 문제를 해결하는 능력을 키울 수 있었다.

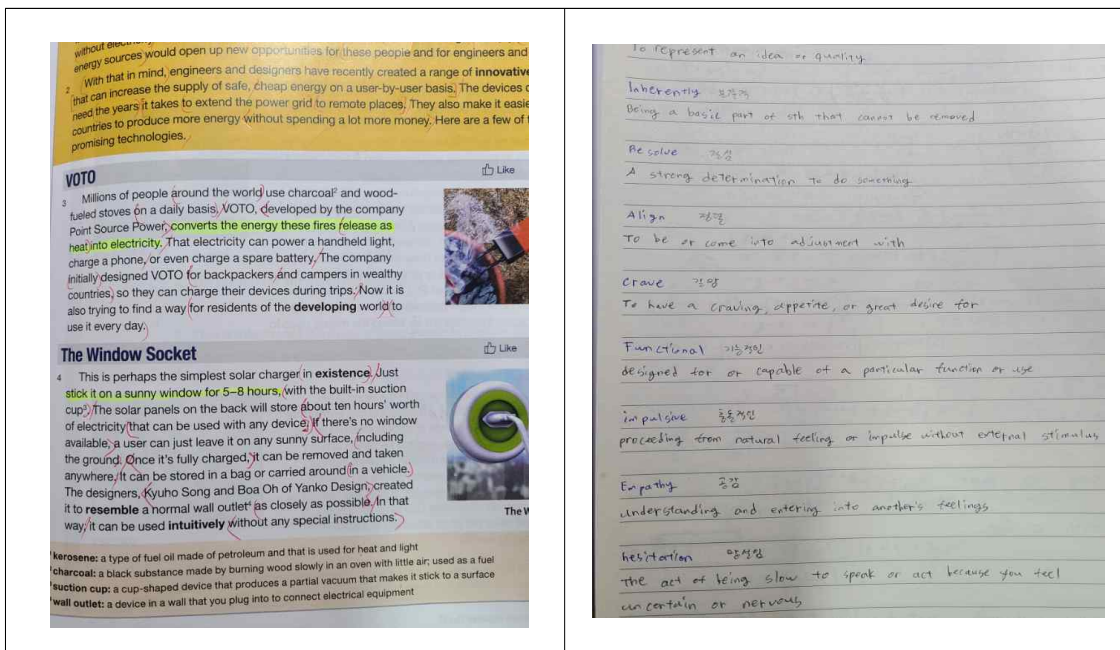
하지만 이 수업이 마냥 학생들끼리만 소통할 수 있었던 것은 아니었다. 교수님은 ‘Pallet’이라는 웹사이트를 따로 만들어 언제나 학생들의 의견을 들어주셨다. 만약 학생이 수업 내용에 이해가 잘되지 않는 부분이 있다면 이 사이트에 접속하여 해당 주차 게시판에 질문사항을 올리기만 하면 됐다. 그럼 교수님이 질문사항을 확인하여 댓글로 답변을 해주시거나, 다음 수업시간에 추가적으로 자세히 설명하여 학생의 이해를 도와주셨다. 이렇게 학생들 간, 그리고 교수와 학생들 간의 다방면적인 소통이 가능했기에 수업성취도 및 만족도를 높일 수 있었다고 생각한다. 그리고 이러한 결과의 배경에는 기본적으로 교수님의 노력이 컸다. 지루할 수 있는 원격수업에서 ‘Kahoot!’이라는 게임 기반 학습플랫폼을 이용하여 학습의 흥미를 유발하고, 단순히 교재에 적힌 내용에서 벗어나 TED 강의와 해외 영화, 팝송 등 다양한 자료를 활용하여 수업을 진행하셨기에 학생들이 적극적으로 수업에 임할 수 있어서 소통도 원활하게 이뤄질 수 있었다.

이처럼 이 수업에서 교수님은 학생들의 적극적인 수업 참여를 유도하시고 서로 만날 수 없는 비대면 환경 속에서도 다양한 방법을 이용하여 함께 소통할 수 있었기에, 그동안 경험해보지 못한 수업 방식으로 접해본 이 강의는 주입식 교육에 익숙해진 나에게 신선한 충격으로 다가왔다. 한 학기 동안 이 강의는 100% 비대면으로 진행되었으나 이미 같은 수강생들이 누구인지 서로 알 정도로 상호 소통 과정이 잘 이뤄져 원격수업의 단점을 전혀 느낄 수 없을 만큼 질 높은 수업을 받을 수 있었다. 끝으로 이러한 상호작용 활동은 자연스럽게 높은 학업성취로 이어졌기에, 이번 공모전을 통해 이런 수업을 만들어주신 교수님께 감사함을 표하며 다른 사람들에게도 이 수업을 추천해드리고 싶다.

2. 나만의 학습 노하우

이 강의는 일주일에 2번씩 이뤄졌으며 각각 동영상 강의와 실시간 강의로 구성되었다. 먼저 동영상 강의로 그 주에 수업할 내용을 설명해주시고, 다음 실시간 수업에서 이 내용을 바탕으로 학생들과 이야기를 나누며 조원 토의 활동을 진행하였다. 따라서 실시간 수업에서 조원들과의 토의 활동이 원활하게 이뤄지기 위해서는 기본적으로 그 전 동영상 강의를 성실히 듣고 해당 내용을 잘 이해해야 했다. 그렇기에

나는 우선 동영상 강의에서 교수님이 설명하시는 내용을 교재에 정리하여 강의 주제를 이해하는 것에 집중하였다. 이때 단지 지금 동영상에서 설명하시는 내용에만 집중하는 것이 아니라, 다음 실시간 수업에서 어떤 점을 이야기하실지 예상해보며 강의를 들었다. 예를 들어 동영상 강의를 걱정 기술에 관한 내용이었다면, 걱정 기술이 적용된 사례에는 무엇이 있는지, 장점과 단점은 무엇인지, 개선할 방향은 무엇인지 등 교재에 적힌 내용에서 한발 더 나아가 실시간 수업에서 다뤄질 내용을 미리 예상하고 정리하여 다음 실시간 수업을 준비하였다. 그 결과 강의 주제에 대한 이해도가 깊어질 수 있어 실시간 강의에서 교수님의 갑작스러운 질문에도 당황하지 않을 수 있었고, 토의 활동에서도 조원들에게 내 생각을 더 분명하게 전달할 수 있었다. 또한 교수님은 ‘Kahoot!’이라는 게임 기반 학습플랫폼으로 영어 단어 시험을 보셨는데, 이 게임에서 고득점을 받기 위해서는 교재에 나온 단어와 예문을 잘 숙지하는 것이 중요했다.



따라서 위 사진처럼 일단 지문에서 모르거나 중요한 단어를 표시하여 따로 정리해 놓고, 이 단어가 쓰이는 예문을 해당 단어 아래 적어 단어의 뜻과 활용을 함께 외었다. 그 결과 실시간으로 진행된 ‘Kahoot!’ 게임에서 정답을 많이 맞힐 수 있었고, 이렇게 다져진 단어는 오랫동안 기억으로 남아 실제 중간, 기말시험에서도 쉽게 정답을 찾을 수 있었다. 이처럼 우선 성실히 동영상 강의를 듣고, 다음 수업시간에서 다뤄질 내용을 미리 준비하는 것이 중요했다. 물론 이것이 학습에 있어서는 가장 기본적인 것일지라도, 이 기본적인 것을 한 학기 동안 꾸준히 실천하는 것이 가장 중요했다.

2등

인지행동치료입문 비대면 수업 참여 후기

교과목명 : 인지행동치료입문

교수명 : 김은하

학과 : 심리학과

이름 : 홍수빈

1. 수업 참여 후기

수업 형식은 매주 월요일에 업로드된 동영상을 시청한 후 학습 보고서를 제출하고, 매주 수요일에 zoom을 통한 실시간 온라인으로 진행된다. 동영상과 실시간을 병행하는 수업이기 때문에 월요일에 동영상으로 배운 수업 내용 중 이해가 안 가는 부분이 있으면 메일로 답을 받거나 실시간 수업이 있는 수요일에 zoom에서 동영상 강의에 대해 교수님께 실시간으로 질의응답을 받을 수 있다.

첫 번째로 인지행동치료 수업에서 의미 있는 상호작용 경험은 토론학습 시간이다. 수요일 실시간 수업 중 수업 중간에 그날 배운 내용들에 대한 토론 주제가 올라오고 10분 가량의 토론 시간이 주어지면 2 ~ 3명의 조원들로 구성된 팀들은 각자 소규모 회의실에 참여하게 된다. 주어진 시간 동안 팀원들은 주제에 대한 토론을 진행하고 토론 시간이 끝나면 다시 회의실로 모여 팀별로 내놓은 의견들을 공유하고 교수님께 피드백을 받는다. 토론학습은 여러 학생들과 의견을 교류하고 토론하고 토론 중 정리한 내용들을 모든 학생에게 공유하도록 격려함으로써 여러 의견들을 듣고 더 깊게 배울 수 있었다. 같은 수업을 듣더라도 이렇게나 다양한 의견들이 나올 수 있음에 감탄했다. 교수님뿐 아니라 토론학습을 통해 학우들에게도 배운 점이 많았다. 특히 토론학습은 수업 중 그냥 지나쳤던 부분들에 대한 재학습이 이루어질 수 있는 시간이기도 하다. 비대면 수업을 하면서 아쉬웠던 부분 중 하나가 여러 학생들과 의견을 교류하고 피드백할 수 있는 공간이나 기회를 얻기 힘들다는 것이었다. 대부분의 수업들이 그러하지만, 특히 비대면 수업 특성상 다른 학우들과 토론학습을 하기란 쉽지 않다. 그러나 인지행동치료입문 수업에서는 토론학습 시간이 따로 주어졌고, 이를 아주 효율적으로 활용하였다는 점이 굉장히 마음에 들었다.

다음으로 인지행동치료입문 수업에서 인상 깊었던 상호작용 경험은 역할연습 과제였다. 역할연습 과제는 실제 심리적 어려움을 겪고 있는 주변인을 찾아 그동안

배운 인지행동치료의 개념들을 적용하여 그의 심리적 문제의 원인을 탐색하고 치료 계획을 세우는 것이다. 첨부된 <이미지 1>은 내가 역할연습을 위해 작성한 활동 계획표다.

- 활동 계획표

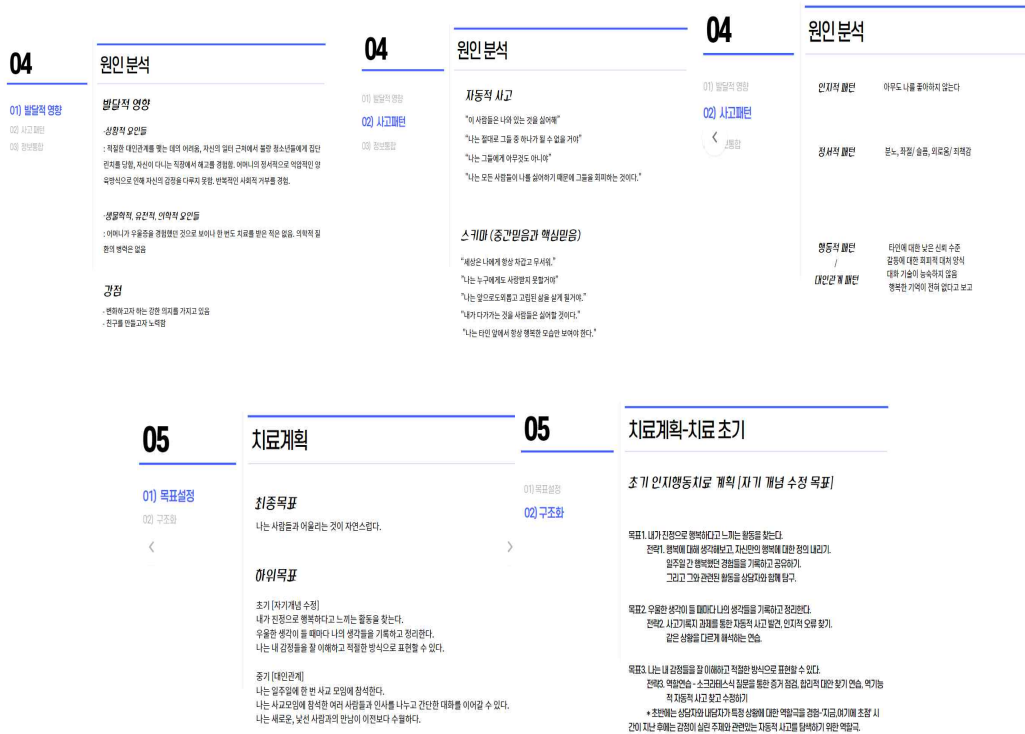
일시	5.1[의제: 문제 탐색]	5.10 [의제: 자동적 사고 탐색]	5.19[의제: 대안적 사고하기 훈련]
	OT(CBT에 대한 설명, 문제에 대한 간단한 질문, 회기에 대한 설명 등 시행)	과제검토(생각기록지 토대로 자동적 사고 탐색)	역할연습 결과물 토대로 인지적 오류, 지지, 반대 증거 찾기
	-문제 탐색 (내담자가 자유롭게 말하도록 격려, 개방형 질문/재진술 통해 내담자의 문제 상황 파악)	자동적 사고 찾기 - 역할 연습 (구체적 사건, 감정, 자동적 사고에 대해 다루기)	-대안적 해석하기 연습(브레인스토밍, 실패에 대한 귀인을 수정) -실패 상황에 대한 행동적 원인& 해결방안 탐색
과제	생각기록지 작성	역할연습 결과물 자기검토	현실적 플래너 작성 연습

<이미지 1> 역할연습 활동 계획표

이론적으로만 배웠던 내용을 실제로 적용하는 과정은 나에게 꽤나 어려운 도전이었다. 내담자의 주 호소 문제를 탐색하고 상담하는 과정에서 그의 핵심믿음, 중간믿음, 자동적 사고를 파악한 후, 그가 가지고 있는 역기능적 스키마를 발견하고 그에 맞는 치료 계획을 세워야 한다. 이 일련의 과정은 인지행동주의적 입장에서 내담자의 문제를 접근하여야 하며, 그러기 위해서는 CBT에 대한 충분한 이해와 숙련이 필요했다. 심리학과 학부생이 상담 기회를 갖는 것은 결코 쉬운 일이 아니다. 그렇기 때문에 수업을 통해 이러한 기회를 접할 수 있었던 건 나에게 굉장히 의미 있는 경험이 되었다. 한 번도 상담 경험이 없던 내가 상담자가 되어 다른 학생을 진지하게 상담해야 하는 과정은 사실 꽤나 어색한 일이다. 그럼에도 이 기회를 통해 상담 과정 전반에 대해 공부할 수 있었고, 상담은 이론적 앎으로만 이루어지지 않는다는 것을 몸소 깨달았다. 내담자는 나의 미숙한 상담 진행에도 최선을 다해 따라와 주었고 상담 진행 동안 제시한 과제 또한 성실히 임해주었다. 내담자의 적극적 태도 덕분에 나의 미숙한 진행에도 불구하고 순조롭게 진행할 수 있었다. 역할연습을 진행하기 전 나름의 준비를 하면서 가장 중요하게 생각했던 점은 내담자에게 온전히 집중하고 그의 말에 경청하는 것이었다. 회기 동안 계속해서 나의 태도에 주의를 가하고 흐트러짐 없는 모습을 유지하기 위해 노력했다. 회기가 끝날 때쯤, 내담자는 나에게 자신의 말을 진지하게 들어줘서 정말 고맙다란 말을 했다. 나의 노력이 내담자에게도 잘 전달되었다는 점에서 역할연습 중 내가 가장 잘한 일이라 생각한다.

마지막은 같은 팀 발표 경험이다. 팀 발표 과제는 같은 수업을 듣는 학우와 한 팀이 되어 CBT의 이론적 개념들을 토대로 심리적 문제를 겪고 있는 가상의 인물에 적용하여 발표하는 것이다. 한달 가량 팀원들과 팀 발표 과제를 준비하면서 상담

이론을 적용할 수 있는 영화 캐릭터를 선정하고, 인적사항, 호소 문제, 인지 과정, 심리적 문제의 원인 분석과 치료 계획, 그리고 실제 적용했을 시 기대효과 등을 설정하였다. 각자 수집한 자료들을 가지고 zoom을 통해 일주일에 두 번의 회의를 거쳐 ppt 공동 작업을 진행했다. 가상의 캐릭터에 CBT 이론을 적용하여 문제 상황을 탐색하고 치료 계획을 세우는 것 또한 인지행동치료를 공부하는 데 매우 많은 도움이 됐다. 특히 사례 개념화를 적용하여 캐릭터가 가지고 있는 문제 상황의 원인에 대해 세세히 분석하는 과정에서 어려움을 겪었지만, 팀원들과 회의를 하면서 서로가 이해한 바를 끊임없이 공유하고 피드백하고 이를 통해 공동작업물을 수정해나가는 과정에서 많은 것들을 배웠다.



<이미지 2> 팀 발표 공동작업물 중 일부

원격수업 참여 사례에 인지행동치료 입문 수업을 선택한 이유는 앞서 언급한 이유에도 있지만, 이 모든 과정에 뒷받침되는 교수님의 열성적인 관심과 피드백 때문이다. 김은하 교수님은 실시간 수업이 끝나면 학생들이 모두 나갈 때까지 퇴장하지 않으시고 학생들의 질문을 기다리고 그에 답해주셨다. 수업이 끝난 후에도 교수님이 느긋한 태도로 남아주신 덕분에 부담 없이 수업 내용에 대해 궁금했던 부분들을 질문할 수 있었다. 학생으로서 교수님과 부담 없이 질의응답 시간을 가질 수 있다는 사실은 매우 큰 가치가 있는 것이다. 수업 내용 중 잘 모르겠거나 모호한 부분

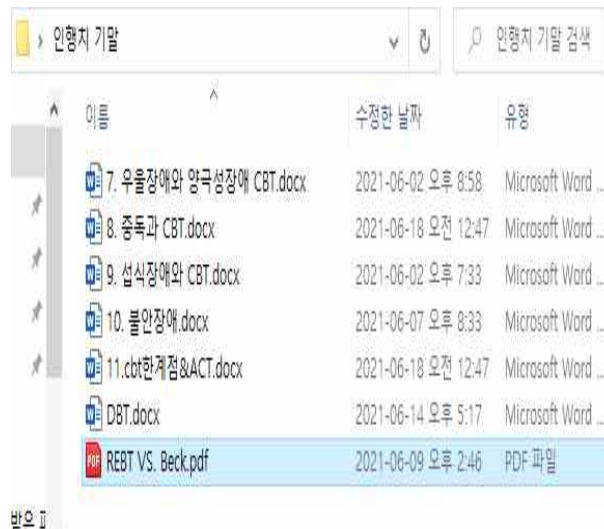
들에 대한 질문을 보류할 경우, 후에 질문했어야 할 거의 대부분의 내용들은 계속 모르는 상태로 남아있게 되는 경우가 일쑤이기 때문이다. 하지만 김은하 교수님 수업에서는 모든 질문에 항상 최선을 다해 답변해주시려 노력하시고, 수업이 끝난 후에도 질문 시간을 만들어주신 덕분에 모르고 넘어가는 부분들 없이 학습을 진행할 수 있었다. 그뿐만 아니라 팀플레이나 개인플레이 및 토론학습 등을 통해 공유되는 학생들의 의견에 적극적으로 피드백해주시고 격려해주셨기 때문에 수업에서 발생하는 모든 학습 과정이 활발히 이루어질 수 있었다고 생각한다.

2. 나만의 학습 노하우

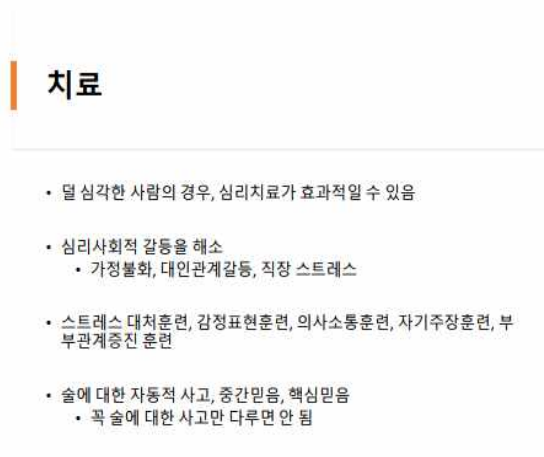
인지행동치료 입문 수업은 인지행동치료(CBT)의 기본 개념과 이론적 배경 및 치료 과정 등을 배우는 이론 중심 수업이다. 따라서 수업의 목표는 인지행동치료의 개념과 이론적 배경을 습득하고, 이를 토대로 인간의 인지, 행동 및 감정에 대한 이해를 높이는 것이다. 이러한 이론 수업에서 가장 중요한 것은 나만의 언어로 재구성한 학습 노트를 만드는 일이라 생각한다. 강의노트를 프린트하고 그 위에 필기하는 것도 좋지만, 가장 학습 효율을 냈던 방법은 강의노트와 강의 내용을 중심으로 워드 파일을 만들어 나만의 필기 노트를 작성한 것이었다. 아무리 좋은 학습 자료라도 나의 손을 거치지 않으면 완전히 내 것이 되기 힘들다고 생각한다. 강의를 들을 때는 모든 내용을 이해했다고 생각하지만, 필기 노트를 만들지 않는다면, 후에 강의노트만으로 기억이 나지 않거나, 막히는 부분들이 많았기 때문이다. 특히 가장 문제가 되는 부분은 강의에 대한 기억이 잘 나지 않을 때 강의노트의 내용을 내 방식대로 왜곡해서 해석하는 것이었다. 이렇게 해석하고 기억한 내용들은 당연하게도 시험에서 좋지 않은 점수를 받았다.

2년 동안 비대면 수업을 경험하면서 터득한 나만의 방법은 다음과 같다. 우선, 수업 시작 전에 강의노트를 다운받고 이를 워드로 옮겨 적는다. 물론 pdf 형식의 강의노트를 워드로 변환하여 저장하는 쉽고 빠른 방법도 있지만, 세부적인 내용들을 직접 타이핑하여 작성하는 작업을 통해 따로 시간내서 연습하지 않아도 연습한 것과 같은 효과를 얻을 수 있다. 최소한 수업 40분 전엔 강의노트를 다운받고 그 주 강의노트를 전체적으로 한번 훑은 후, 해당 주제 배우게 될 내용들의 굵직한 맥락들을 파악한다. 이 과정에서는 세세한 내용들까지 파악할 필요는 전혀 없다. 이렇게 전체적인 맥락을 파악했으면, 워드에 제목과 소제목들을 적는다. 그 후에 해당 제목에 들어갈 내용들을 직접 타이핑하면서 다시 한번 내용들을 훑어본다. 이때 잘 읽히지 않는 부분들은 하이라이트 표시를 하고 넘어간다. 이렇게 작성된 문서 파일을 가지고 수업에 참여한다. 수업 중 교수님이 설명한 내용들은 작성된 강의노트 내용

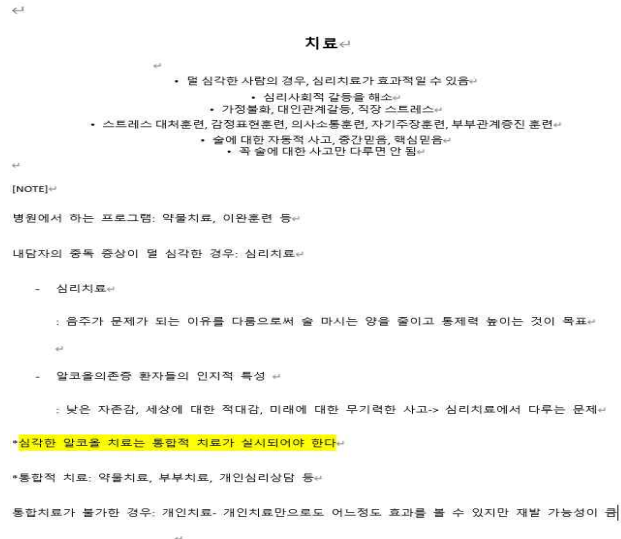
밑에 [NOTE]란을 따로 만들어 타이핑하고, 수업이 끝난 후에 다시 전체적 맥락을 따라 훑어보면서 복습한다. 복습 과정에서 이탈자나 서식에 어긋나는 부분들을 찾아 바로바로 수정한다면 후에 시험 준비를 할 때 필기 노트를 만드는 데 많은 시간을 들이지 않아도 되기 때문에 굉장히 효율적으로 학습할 수 있다. 이렇게 수업 전 40분, 수업 직후 30분 정도의 시간을 투자한다면, 따로 시간을 들여 예습, 복습을 하지 않아도 되기 때문에 의지가 약한 학생들도 얼마든지 쉽게 접근할 수 있는 방법이라 생각한다. 수업 2주 전까지는 수업을 듣고 이와 같은 방법으로 필기 노트를 만들어 미루지 않고 예습, 복습하는 것으로 학습을 진행하는 것이 효과적이다. 부차적으로 어려웠던 부분이나 잘 이해가 가지 않는 부분은 교재와 질문을 통해 그 자리에서 해결하고, 이 과정에서 새롭게 얻은 지식은 필기 노트에 꼼꼼히 적어둔다.



<이미지 3> 인지행동치료 필기 노트 파일



<이미지 4-1> 강의노트 중 일부



<이미지 4-2> 필기 노트 중 일부

위에 첨부된 <이미지 3>은 지난 학기 동안 만든 인지행동치료 기말고사 범위 필기 노트 정리 파일이며, <이미지 4-1>은 강의노트, <이미지 4-2>는 첨부된 강의노트에 해당하는 필기 노트다.

이후 시험 준비 기간에는 이렇게 완성된 필기 노트들을 복기하고 필기 노트의 내용들을 계속해서 정리한다. 시험 1주 전에는 지금까지 만든 필기 노트들을 토대로 하나의 워드 파일을 만들어 모든 내용들을 서식에 맞게 정리하여 넣은 후, 최대한 중심 내용들을 토대로 내용들을 정리하여 프린트하고 가지고 다니면서 틈틈이 학습한 후, 시험 범위에 해당하는 챕터들 각각의 백지 노트를 만들어 범주화한 후, 해당 범주에 해당하는 내용들을 기억나는 대로 적은 후, 필기 노트를 토대로 이전에 작성한 백지 노트를 수정한다. 이 작업을 누락된 부분이 없을 정도로 반복하는 것이 완벽히 학습하고 기억할 수 있는 방법이다.

이러한 일련의 과정은 PQ5R 학습법과 관련이 있다. PQ5R 학습법이란 인지 행동적 원리를 기반으로 만들어진 효율적인 학습 전략으로서 학습 전, 학습 후, 전략적 반복이라는 세 단계에 걸쳐 진행된다. 첫 번째로 학습 전 단계는 Preview와 Question하는 단계이다. 전체 장의 구성을 훑으면서 전체 장에 포함되어있는 하위단락들과 소제목을 살펴본다. Preview 과정을 통해 공부할 내용에 어떤 부분들이 들어있는지 파악하는 일종의 틀을 만든 후, Question으로 넘어간다. 전체적인 내용을 개관한 후에 떠오르는 생각과 의문을 정리하면서 질문 리스트를 만드는 단계이다. 학습 전 단계가 끝나면 학습 중&후 단계로 넘어간다. 학습 중&후 단계는 Read, Recite, Reflect, Review하는 단계이다. 꼼꼼하게 문장 하나하나를 읽으며 학습 전 단계에서 생긴 의문들의 답을 찾는 과정이다. 이를 토대로 새로 접한 개념들을 암기하고, 읽기가 끝났다면 내용을 전체적으로 숙고하고 구조를 머릿속에 떠올리며,

반복적으로 학습하면서 학습한 내용들을 조직적으로 정리한다. 정리가 끝났다면 마지막으로 전략적 반복 단계로 넘어간다. 전략적 반복 단계는 Retrieval Practice하는 단계로 이 과정에서 인출 연습을 한다. 말 그대로 지금까지 학습한 내용들을 머릿속에서 끄집어내는 연습을 하는 것이다.

앞서 나만의 학습법에서 언급한 훑어보기, 의문 제기, 읽기, 반추하기 과정은 강의노트를 개관하고 필기 노트로 옮겨 작성하는 과정에서 경험하는 인지 과정에 해당하며, 후에 필기 노트를 통해 학습하는 과정은 암기하기, 복습하기에 해당한다. 마지막으로 시험 전에 반복하는 백지 노트는 인출 연습 전략으로 사용할 수 있다. 이러한 인지 과정은 배운 내용들에 대한 확실한 이해 암기뿐 아니라 학습한 내용들을 자유자재로 사용할 수 있도록 도와준다. 뿐만 아니라 필기 노트를 자가 첨삭하는 과정에서 이차평가가 이루어지며, 이 과정을 통해 자신의 학습이 잘 이루어지고 있는지를 알 수 있다.

나는 실제 이 학습법을 사용한 대부분의 시험에서 우수한 성적을 받았으며, 필기 노트를 작성하고, 복기하고, 백지 노트를 작성하는 학습법에 익숙해지는 것은 대학 시험뿐만 아니라 대부분의 학습 장면에서 좋은 성과를 얻는 방법이 될 수 있으리라 생각한다.

3등

현대자동차, 그리고 UAM - <전략기술경영> 강의를 듣고

교과목명 : 전략기술경영

교수명 : 이성주

학과 : 산업공학과

이름 : 김서란

1. 수업 참여 후기

‘팀플이 많아, 근데 재밌어.’ 이전에 전략기술경영 강의를 들은 선배님들이 입을 모아 해준 말이었습니다. 저희 학과는 학기마다 굵직한 팀 프로젝트를 수행하고 발표해야 하는 강의가 많았습니다. 저는 이런 팀 프로젝트가 싫지만은 않았습니다. 학교를 다닌 지난 4년간 팀 프로젝트만큼 뚜렷한 결과물과 성취감을 만들어내는 활동은 없었기 때문입니다. 또 이론적으로 배운 내용을 근거로 하여 실제로 내가 원하는 주제에 적용을 하는 것만큼 확실한 배움은 없다고 생각합니다. 협력을 하며 미처 생각하지 못한 부분을 짚어주는 팀원들과 팀에 대한 책임감이 저를 여기까지 만들어주었다고 생각합니다. 밤을 새우고 의견을 조율하는 과정은 힘들었을지언정 저에게 남겨진 많은 보고서와 PPT들은 큰 자산이 되었습니다.

때문에 4-2, 마지막 학기에도 팀 프로젝트가 있는 수업을 수강하고 싶은 욕심이 있었습니다. 그렇게 제가 전공선택 과목으로서 수행하게 된 팀 프로젝트는 전략기술경영의 설계과제였습니다. ‘선택한 기업의 전략 로드맵을 만들어보는 것’이 강의의 큰 목표였습니다. 그러기 위해선 사전 이론과 기업의 조사 및 분석이 필요했습니다.

중간고사 이전까지는 동영상 중점으로 공부하고 Zoom을 이용한 화상회의를 통해 팀원들과 기업 선택과 분석에 대한 회의를 일주일에 한 번씩 진행하였습니다. 격주로 주어지는 설계과제를 수행하기 위함이었습니다. 이때 저는 현대자동차가 새롭게 시작하는 사업인 UAM(도심 항공 모빌리티)를 선택하였습니다. 평소 좋아하는 기업에서 진행하는, 관심이 가는 미래의 신사업이었기 때문입니다.

중간고사 이후부터는 동영상 강의와 실시간 강의를 병행되었습니다. 기업이 사업과제를 선택하고 추진하는 다양한 방법론에 관한 수업을 듣고 동시에 현대자동차가

UAM 사업을 이끌기 위해서 어떻게 해야 할지 설계하는 과제를 팀별로 수행하였습니다. 그리고 과제를 수행하면서 떠올리는 질문들을 정리해 실시간 강의를 통해 질문을 하였습니다. 다른 학생들의 질문과 해답을 듣고 다른 조의 수행 과제에 대한 피드백을 보며 미처 생각지 못했던 부분에 대한 보충도 가능했습니다. 추가로 교수님께서서는 각 조마다 소회의방을 만들어 화상회의를 진행하게 하시고, 직접 소회의방에 들어오셔서 조별로 질문을 직접 받아주시기도 하셨습니다. 특히 저희 조는 현대자동차가 국내 선두주자로서 시행하고 있는 신사업이기 때문에 적용과 판단이 어려운 점이 많았는데 이때 교수님과의 많은 대화를 통해 수월히 진행할 수 있었습니다.

‘팀플이 많은데 비대면으로 어떻게 진행해?’에 대한 걱정은 전혀 들지 않았습니다. Zoom 회의를 활용해 저의 화면을 보여주면서 의견을 나누었습니다. 특히 주제를 선정할 때 팀원들에게 제가 원하는 현대자동차의 UAM을 설득하기 위해 간단한 PPT를 제작해 보여주었고, 회의 주제가 여러 가지일 때 PPT에 안전을 정리하여 보여주고 그 PPT로 바로 각자 해야 할 일을 정리하였습니다. 급하게 회의를 해야 할 때는 바로 Zoom 회의방을 만들어 진행하였습니다. 시간을 내고 준비해 만나는 것보다 훨씬 더 효율적이고 안전하였습니다. 앞으로 대면으로 수업이 전환되어도 저뿐만 아니라 다른 학우분들까지도 비대면 회의를 애용할 것으로 보입니다.

이렇게 프로젝트를 진행한 후 학기 말에는 지금까지 배운 것들을 총정리하는 최종 설계과제에 대한 발표가 있었습니다. 궁극적으로 ‘현대자동차가 UAM 사업을 진행하기 위해선 어떤 전략을 어떤 시기에 어떻게 진행하는 것이 좋겠다.’에 대한 결론을 내리는 내용이었습니다. 교수님과 학우분들이 현대자동차 UAM사업부장이라고 생각을 하고, 그들을 설득하기 위한 방향으로 PPT를 제작하였습니다. 특히 데이터 분석, 파이썬을 이용한 topic modeling 등의 기술적인 면을 최소한으로 줄이고 사전 지식이 없는 사람들까지 이해할 수 있는 방향으로 준비하였습니다. 이것이 교수님께서 원하시는 방향이라고 생각하였기 때문입니다.

중간고사와 기말고사 또한 단순히 외우는 암기 형식이 아닌 자신의 생각을 서술하는 방식이었습니다. 대기업과 관련된 뉴스가 제시되면 이에 대한 생각과 이런 경우 어떤 판단을 하는 것이 좋을지 배운 내용을 근거로 적는 방식이었습니다. 비대면 시험의 단점 중 하나는 부정행위의 가능성이 큰 것인데, 이 경우에는 오픈북에다가 본인의 생각을 적는 것이므로 베끼는 것의 의미가 없었습니다. 이해를 하고 적용을 할 줄 알아야 했습니다. 시험을 대비하기 위해 현대자동차의 UAM 사업부만의 관점이 아닌 다양한 기업의 관점에서 생각해보아야 했습니다. 때문에 기업들의 기술 개발 동향에 관한 뉴스들도 직접 찾아보고 팀원들과 이야기를 해보기도 하였습니다.

평소에 뉴스에 관심이 적었던 제가 저절로 이렇게 찾아보게 된 것은 교수님의 수

업 방식 덕이기도 하였습니다. 일반적으로 지식을 전달해주시기보다는 매 수업마다 open case로 기업의 사례를 보여주시고 질문을 던지셨습니다. 저는 강의를 듣는 내내 그 질문에 대해 답을 하려고 애를 썼습니다. 질문에 focus를 맞춘 채 듣는 수업은 그냥 받아들이는 것보다 확실히 남달랐습니다. 강의의 마지막에는 discussion questions이 실려 있었고 이를 통해 배운 내용을 간단하게 훑어주셨습니다. 이 discussion questions의 유형과 비슷하게 중간고사, 기말고사에 출제가 됩니다. 이 사례들과 질문들을 바탕으로 최근 LG와 SK간의 배터리 분쟁 등의 다양한 이슈에 적용해보고 싶어 스스로 기사를 찾아보고 팀원들과 이야기를 나누었습니다.

동영상을 통해 배운 이론을 팀원들과 함께 선택한 기업과 사업에 적용해보고, 스스로 다른 기업의 상황에서도 적용해보며 완벽히 체득하였습니다. 교수님과의 소통을 통해 방향성을 확인하고 다른 학우분들의 설계과제 피드백을 보며 미처 생각하지 못한 부분까지 확인하였습니다. 나와 팀원들 - 다른 수강생들 - 교수님. 이렇게 삼각형 구조의 소통이 완벽했던 강의였습니다. 2020-1학기, 2020-2학기에 이어 3번째 비대면으로 이루어지는 강의로 모두가 익숙해진 상태였기 때문에 화상회의 등의 이용이 적극적으로 이루어져 평소 대면 때보다 훨씬 상호작용이 더 잘 된 케이스라고 생각합니다.

2. 나만의 학습 노하우

① 기초가 제일 중요하다, 개념을 확실히 정리하기

단순 암기가 필요 없는 시험이었지만 개념은 무조건 짚고 넘어가야 했습니다. 개념에 대한 이해가 바탕이 되어야지 여러 응용을 할 수 있기 때문입니다. 저는 평소 이해가 안 가는 개념이 있으면 다시 맨 앞 장으로 돌아가 정의를 꼼꼼히 익히곤 합니다. 모든 것들은 정의부터 시작됩니다. 너무 뻘하고 모든 과목에 적용되는 노하우이지만 그만큼 빼놓을 수 없는 것이라고 생각합니다.

② 이해한 것을 바탕으로 내 언어로 정리하고 바꾸기

강의노트에 꼼꼼히 필기를 한 후 자기주도학습을 통해 논리적으로 현상과 개념을 이해해야 합니다. 가령 기술의 표준전쟁에서 기업이 승리자가 되기 위해서는 이런 것들이 필요하다는 것을 배웠을 때, 표준은 왜 만들어지며 모든 산업에서 표준이 생길까? 현재 2차 전지 사업에서 시장점유율이 가장 높은 그룹은 LG인데 최근 중

국의 CATL에게 1위를 빼앗겼다, 무슨 이유일까? 까지 사고를 확장시켰습니다. ‘질문에 대답할 순 있을 것 같아!’ 정도에서 그치면 안됩니다. 면접에서 질문이 나오면 대답을 할 수 있을 정도로 나만의 언어로 정리하고 교수님께 질문을 통해 확인하면 이것이야말로 완벽한 나의 것이 됩니다. 따라서 논리적인 이해가 가장 중요하다고 생각합니다.

③ 한글 용어뿐만 아니라 영어 용어까지 알기

영어로 된 강의노트를 한글로 직접 설명해드립니다. 신입생 때에는 한글로 설명해주실 걸 왜 어려운 영어로 굳이 강의노트를 구성하셨을까 의문이 들었습니다. 이는 자연스럽게 영어 실력을 향상시켜 줄 뿐만 아니라 영어 용어가 익숙해져야 한다는 교수님들의 큰 뜻이라는 걸 뒤늦게 깨달았습니다. 파파고 번역기가 해석해준 대로 아는 것은 무의미합니다. 다른 사람과 효율적인 의사소통을 하기 위해서는 특정한 영어 용어를 제대로 아는 것이 중요하다는 것을 느꼈습니다. 때문에 초반에 용어를 처음 접할 때는 한글 용어를 바탕으로 이해하되 확실한 용어에 대한 암기는 영어로 하는 것이 좋습니다.

④ 팀원들과 함께 공부하기

다른 과목에서도 마음이 맞는 친구들과 함께 팀을 구성하였을 때에는 매우 효과적인 방법일 것이라고 생각합니다. 특히 이 강의에서는 매주 공부한 내용을 팀원들과 함께 현대자동차의 UAM 사업에 적용해야 했기 때문에 더욱 효과적이었습니다.

팀원들과의 Zoom 회의 시간 전까지 강의를 필수적으로 듣고, 현대에 적용시키려면 어떻게 해야 할까 까지 생각해보는 것이 매주 해야 할 일이었습니다. 때문에 저절로 스터디가 가능했습니다. 어려웠던 부분에 대해서 팀원은 어떻게 이해했는지 확인하고, 교수님께서 강의에서 던지신 open case에 대한 질문들과 맨 마지막 discussion 질문들에 대한 답도 함께 정리해봅니다. 이는 저절로 중간고사, 기말고사 공부까지 이어집니다. 가장 효과적인 학습 노하우이며 이것이 결국 나와 팀원들 - 다른 수강생들 - 교수님 간의 삼각형 구조의 소통을 이끌어낼 수 있었던 주요 요인이었던 것 같습니다.

졸업을 앞둔 바쁜 마지막 학기였음에도 불구하고, 학기를 진행하는 내내 ‘정말 듣기 잘했다!’ 라는 생각이 절로 드는 강의였습니다. 이 강의를 듣지 않고 졸업했다면 정말 아쉬울 뻔 했습니다. 산업공학 학부 과정을 모두 배운 후 경영진의 입장에서 기업의 전략을 세우는 경험은 지금까지 해왔던 연구진·팀원 입장에서의 ‘효율

적으로 일하기' 를 주제로 한 다른 팀 프로젝트와는 사뭇 달랐습니다. 산업공학과
의 후배분들은 물론 경영학과 분들과 마케팅에 관심이 많으신 분들까지 강력 추천
하는 강의입니다.

이렇게 만족도 높은 상태에서 즐겁게 학습하고 마무리까지 A+을 받을 수 있었던
저희 노하우까지 소개하며 긴 글을 마치겠습니다. 즐겁게 들었던 강의의 강의노트
도 다시 확인해보고, 수행했던 과제들과 최종 PPT까지 기분 좋게 되짚어볼 수 있었
습니다. 감사합니다.

3등

한국근세사 비대면 수업 참여 공모

교과목명 : 한국근세사

교수명 : 한상우

학과 : 사학과

이름 : 문현진

1. 수업 참여 후기

<상호작용 사례>

(1) 교수님-학생

1. 설문조사 : 이 수업의 실시간 수업 프로그램인 줌 기능에 있는 설문조사 프로그램을 사용하셔서 수업 방향, 과제, 수업 내용의 개인 견해 등에 대해 학생 개개인의 의견을 최대한 반영하셔서 수업을 해주셨습니다. 교수님께서 설문조사를 하실 때 “이 수업은 여러분들을 위한 시간이 되도록 힘을 쓰겠다.” 라고 말씀하셨고 학생들에게 좀 더 도움이 될 수 있는 시간을 만들고자 하셨습니다. 설문조사가 끝나면 교수님께서 비밀로 진행을 하셨지만 선택사항으로 왜 그 항목을 선택을 했는지 개인적으로 선택 사유를 발표할 수 있는 시간을 주셨습니다.

예를 들면 첫째, 수업 방향은 선택 사유에 대해 학생들의 목소리가 나오면 그 내용을 참고하셔서 교수님께서 원래 계획하셨던 계획을 수정하셔서 아주Bb에 공지해 주셨습니다. 이후에는 공지해 주신대로 수업을 해주셨습니다.

둘째, 과제는 총 2개 있었는데 조별과제는 조선시대의 역사적 사건 중 논쟁이 되는 주제로 발표하는 것과 2인 1조의 조 편성 선호도에 대한 설문조사를 하셨습니다. 개별과제는 개인 가족사에 대한 서술을 하는 것에 대한 설문조사였습니다. 원래는 집안의 족보, 가보와 어르신들의 증언을 중심으로 작성을 하는 것이라고 안내를 해주셨습니다. 그런데 저를 포함하여 일부 학우님들은 말하기 어려운 개인 사정이 있어서 과제 작성에 대한 우려가 되는 점을 교수님께 말씀드렸습니다. 교수님께서 사정을 이해해주셔서 사정이 있는 학생은 사정을 개별적으로 전부 다 들어주신 다 음에 최대한 작성을 할 수 있도록 많은 조언을 해주셨습니다.

셋째, 수업 내용의 개인 견해는 교수님께서 수업을 하신 다음에 교수님과 학생이

서로 배워가자는 의미에서 수업 내용에 대한 찬성과 반대, 여러분이 당사자였다면? 등의 주제로 설문조사를 하셨습니다. 설문조사 결과가 나오면 발표할 수 있는 기회를 주셨습니다. 자발적으로 저와 학우님들이 발표를 하면 교수님과 토론을 하면서 저 개인적으로 다양한 의견을 공유할 수 있는 시간이라서 수업 내용에 대해 더 많은 주제로 생각을 해보고 수업 내용을 이해하는 데 많은 도움을 받았습니다.

2. 수업 종료 후 질문의 시간 : 매 실시간 수업마다 교수님께서 수업을 마치시면 질문이 있는 학생은 줌 나가기를 하지 않고 남아서 줌 마이크를 통해 질문을 할 수 있는 시간을 주셨습니다. 이 시간에 저는 수업 내용 중에 이해가 필요한 대목과 효율적인 과제를 위한 계획에 대해 질문을 드렸습니다. 다른 학우님들도 이러한 내용으로 질문을 드렸습니다. 교수님께서서는 질문을 주셔서 감사하다고 인사를 해주시면서 기분 좋게 답변을 해주셨습니다.

3. 메일 : 교수님께서서는 수업이 없는 날에도 수업에 관련한 질문이 있으면 무엇이든 메일로 보내면 답장으로 설명해주신다고 하셨습니다. 저는 교수님께 과제 진행도에 대해서 메일을 드린 적이 있었는데, 교수님께서 진행도는 적절하고 보완할 점에 대한 걱려가 담긴 조언의 메일을 보내 주셨습니다.

4. 동영상 강의 : 이 수업은 중간고사를 실시하지 않았습니다. 교수님께서 학생들의 다른 과목 시험공부를 위하여 중간고사가 다가오는 날의 수업은 동영상 녹화 강의를 아주Bb에 올려주셔서 시간에 구애받지 않으라고 배려도 해주셨습니다.

(2) 학생-학생

- 조별과제의 조원과의 소통 : 위 (1)의 둘째에서 언급한 2인 1조 조별과제를 위해 교수님께서 비대면 수업이라 조원들이 만나기가 어려워서 수업 끝나기 10분 정도 남기고 조원들끼리 묶어서 줌 소회의실을 만들어주셨습니다. 이는 3주 정도 진행이 되었는데 이 시간 동안 저의 같은 조원인 학우님과 주제 선정, 조사, 역할 분담 등의 내용으로 소통을 하였습니다. 하지 못한 대화는 서로 연락처를 공유하여 전화, 문자를 주고받으면서 비대면 상황이었어도 어려움 없이 소통을 하면서 준비했기 때문에 문제없이 발표를 할 수 있었습니다.

- 발표 후 질문 및 토론 : 조별 발표가 끝나면 토론을 하는 시간이 있었습니다. 줌 마이크나 채팅을 통하여 이루어졌습니다. 예를 들면 제가 질문을 드리면, 해당 조원분들이 답변을 해주시면 이에 대해 토론이 이루어지고 여기에 다른 의견이 있는 조가 있다면 참여해서 다른 주장을 밝히는 흐름이었습니다.

2. 나만의 학습 노하우

1. 학습 방법 : 교수님께서서는 암기 위주가 아니라 한국근세사 범위에 포함되는 획기적인 사건을 중심으로 학생 입장에서 자유로운 의견을 내서 서로 토론하는 형식의 수업을 하셨습니다. 다음 수업을 예고해주시면 저는 다음에는 어떤 의견을 내야 할지 준비를 하고 나서 수업에 참여하였습니다. 제 개인적으로 다른 학우님이 저보다 더 좋은 주장을 했다고 느낄 때는 위축되지 않고 배우자는 자세로 수업이 끝나면 그분이 제시했던 의견을 참고하여 저의 주장하는 능력이 향상될 수 있도록 참고하였습니다. 어려운 내용은 위에서 언급한 교수님과 상호작용의 방법(질문 시간, 메일)을 통해 질문을 드리고 답변을 받으면서 이해를 했습니다.

2. 수업 듣기 : 비대면 수업이라고 해서 나태한 생각 없이 제 방의 책상에 있는 컴퓨터가 학교라고 생각하며 한 번도 늦지 않고 제시간보다 일찍 실시간 줌에 접속했습니다. 접속 후에 수업이 시작되기 전까지 지난번의 수업 내용을 되새겼습니다. 동영상 녹화 강의도 실시간 수업이면 화요일, 금요일 C교시였는데 절대 다음 날로 미루지 않고 이 시간에 맞춰서 수강을 했습니다. 녹화 강의도 나중에 수강하는 것은 할 일이 쌓이게 되어 더 바빠지기 때문에 원래 학기 루틴대로 제시간에 수강을 하는 것이 좋았던 것 같습니다.

3. 수업 내용 촬영 및 녹음 : 교수님께서 수업 내용을 파워포인트로 제작하셔서 줌 화면에 띄우시며 원격수업을 하셨습니다. 혹시나 제가 파워포인트의 내용을 필기하다가 다음 페이지로 넘어가시면 해당 페이지 수업 내용을 놓칠 수도 있을 것 같다는 생각이 들었습니다. 저는 첫 수업 이후부터는 교수님께서 파워포인트 화면을 넘기시면 바로 스마트폰으로 사진을 찍었습니다. 그리고 수업이 끝나면 개인적으로 사용하는 스마트폰 메모장에 진도별로 분류하여 저장하였습니다. 교수님께서 파워포인트를 중심으로 설명을 해주시지만 핵심 내용을 강조하시거나 이해를 위한 부연설명을 해주시면 제 스마트폰 녹음기로 녹음도 했습니다.

4. 노트 필기 전략 : 주차와 날짜를 표기하여 당일 원격수업을 수강할 때마다 강의내용과 교수님의 말씀을 노트에 필기를 해왔습니다. 실시간으로 필기를 하다 보니 정리가 잘 되지 않았던 적도 많았습니다. 그래서 위 3번에서 언급한 사진과 녹음 파일을 참고하여 복습할 때 보기 좋게 더 깔끔하게 요약하고 정리했습니다.

5. 시험 대비 전략 : 수업 내용에 대한 개인 견해와 주장을 내세우는 수업 흐름이라 저 개인적으로는 특정 주제를 시험문제로 제시하셔서 그것에 대해 논술을 하는 문제를 출제하실 것 같다고 생각했습니다. 이 예상에 맞춰 그동안 수업시간에 설문 조사와 토론 시간에 제가 말했던 것들과 교수님께서 해주신 수업 내용을 바탕으로 개인적으로 예상 논술 주제를 만들어보고 답안을 작성하면서 기말고사를 준비했습니다. 실제로 교수님께서 논술 형식으로 기말고사를 출제하신다고 안내하셨고, 실제 시험문제도 문제에 대한 개인 주장을 밝히는 논술 문제였습니다.

3등

조직변화 수업으로 ‘나’를 변화시키는 경영의 미학

교과목명 : 조직변화

교수명 : 이태진

학과 : e-비즈니스학과

이름 : 신선아

1. 수업 참여 후기

‘조직 변화(OC)’ 수업은 나를 이루고 있는 것들을 어떻게 경영할 수 있는지 고민할 수 있는 기회로 다가왔다. 아주대학교 선배이신 이태진 교수님의 수업은, ‘조직 변화’의 경영학을 가르침과 동시에 경영학의 주체를 ‘나’로 설정하는 법을 배울 수 있는 수업이었다. 예컨대, 조직이 이루고 있는 ‘MISSION(미션), VISION(비전), 핵심 가치’라는 가치 체계의 기반과 그 도출법에 대해서 학습한 후에는, ‘나를 이루고 있는 가치 체계는 무엇인가?’라는 질문을 던지셨고, ‘창의’와 ‘도전’을 기반으로 한 기업의 창조적 조직문화를 다룬 후에는, ‘내가 주체가 되어 이루었던 창의와 도전의 경험은 무엇인가?’에 대한 물음을 던지셨다. 이러한 질문들을 통해 나는 사진, 패션, 건축 등의 다양한 심미적 활동에 주목하며 예술 경영활동을 점진적으로 전개해왔던 스스로를 돌아보고, 스스로가 레퍼런스(Reference)가 되기 위한 창작 활동과 가치에 대한 통찰을 지속하는 기반을 마련할 수 있었다.

최근 온라인 플랫폼을 활용한 마케팅 방식이 주요한 마케팅 수단으로 활성화되면서, 모든 취향이 비슷하게 공유되고 있으며, 이에 소비자들은 취향의 독창성을 잃고, 포화상태의 기성품에 대한 매너리즘(mannerism)¹⁾에 빠지고 있다는 생각에 닿게 되었다. 나아가 이전의 장르 안에서 비슷하게 회귀하고 모방하며 주류, 비주류의 경계가 모호해지는 패션산업으로 인해, 포화상태의 기성품에 기대어 자신 스스로를 표현하고 있다는 것에 많은 사람들이 불충분함을 느끼고 있다는 이슈를 파악할 수 있었다.

이러한 트렌드에 대한 인사이트를 가졌던 나에게 있어 ‘조직변화’ 수업에서 던

1) 매너리즘은 ‘항상 틀에 박힌 일정한 방식이나 태도를 취함으로써 신선미와 독창성을 잃는 일’을 가리킨다. ‘타성’(惰性)이라는 말과 닮은 뜻이다.

져 주신 질문들은, 당면한 문제를 해결하기 위해, 이에 대한 역풍으로 ‘다름’을 추구하고 갈망하는 소비자들로 하여금 아름답다고 느끼는 것들을 주체적으로 향유하기 위한 새로운 경향을 만들어내고 싶다는 생각을 할 수 있게 했다.

따라서 수업을 통해 이러한 이슈를 비즈니스의 관점에서 해결하기 위한 가상의 기업을 고안하여, 수업을 통해 배운 ‘가치 관리’를 적용한 [가치 관리 종합 - 가상 기업 설립을 통한 가치 관리 고찰] 보고서를 통해 DIY 패션 브랜드인 ‘DRESTE’를 설립할 수 있었다. 이때, ‘DRESTE’는 기업의 디자이너들이 기본적인 옷의 도면을 다양한 장르와 웨입(shape)으로 제작하고, 소비자는 사이트 내의 툴을 이용하여 자신만의 옷을 커스텀 한 뒤 배송받는 시스템으로 운영된다. 따라서 브랜드의 미션(MISSION)은 ‘다름을 추구하는 독보적인 디자인과 혁신적인 경영시스템으로 최고의 제품과 서비스를 창출하여, 소비자들이 주체적으로 각자의 취향과 개성을 인식하고, 향유할 수 있게 돕는다.’로 설정하였고, 비전(VISSION)은 ‘대한민국에서 가장 독보적인 의류 디자인 업체’로 설정하였다.

이후 사회적 가치와 경제적 가치를 동시에 추구하는 공유가치창출(CSV)개념을 배운 뒤, 위와 같이 설립한 가상 기업을 CSV의 관점으로 경영전략을 재수립하여 수업 내용의 간 학문적인 확장을 주체적으로 도모할 수 있었다. 가상 기업의 핵심역량은 개인이 가진 다름을 인정하고, 추구하는 브랜드로써 이전의 기성품을 향유하던 방식에서, DIY로 고객의 옷을 직접 커스텀한다는 점에 있음을 분석하였다. 즉, 기능 가치, 사용가치에서 나아가 인지적, 감정적 공감을 기반으로 고객의 가치관과 생활에 변화를 주는 심리적 가치(Value)단계인 정서 가치를 추구하고 있음을 통찰하여, 사회적인 문제로 비즈니스의 가능성을 확장했다. 패션업계는 그동안 비장애인을 위한 옷과 디자인에 집중한 반면 다양성을 반영하지 못하는 경향이 지배적이었으며, 장애인은 기성복의 불편함을 해소할 수 없었고, 의류 선택권을 보장받지 못했던 실정을 파악했다. 이러한 문제를 해결하기 위하여 공유가치창출 전략을 방법론적으로 접근해보았을 때, 이해관계자에 대한 유형을 ‘신체장애인을 비롯해 신체적 차이로 기성복 소비에 불편함을 겪는 고객으로 분석하고, 최근 장애인의 패션 권리를 강조하여 개인의 개성이 드러날 수 있도록 옷을 디자인하는 어댑티브 패션(adaptive fashion)’이 패션의 한 카테고리로서 부상되고 있는 경향에 주목하였다. 이에 기업의 미션을 공유가치창출(CSV)의 관점에서, ‘다름을 추구하는 독보적인 디자인과 혁신적인 경영시스템으로, 개인의 신체적 특성과 관계없이 패션의 권리를 보장하고, 누구나 개인의 취향을 향유하고, 심미성을 추구할 수 있도록 돕는다.’로 재정의할 수 있었다. 즉 사회적 니즈와 이익을 충족시키기 위한 제품과 시장의 기회를 다시 찾고, 제품이나 소비자 차원에서 접근하여 재인식함으로써 사회적 이슈를 찾아 대응하여 생산성을 재정의하는 과정을 직접 경험할 수 있었던 것이다.

나아가 이러한 과정은 조직의 고유 미션(존재가치)을 강화하여 주요사업이 지속발

전 가능한 사업이 될 수 있도록 지속적인 재검토가 필요함을 인지하고 클러스터 차원에서 접근하여 지역산업 클러스터의 개발을 통해 재활의학과 전문의, 재활공학서비스 연구지원센터, 정부의 장애인 대상 의류 사업 지원금 지원 등, 정부, 연구/교육기관, 사회 기반시설 등의 지역 산업 클러스터의 새로운 개발과 구축을 통한 공유가치 창출 전략을 실천 방안을 설립할 수 있었다.

이처럼 공유가치창출(CSV) 관점의 전략 수립 방법을 실제 사례에 대입하여 가치체계를 재정립하고, 과정에서의 고찰을 통하여 공유가치창출의 의의를 미개척 사업 분야에 대한 시장 가능성을 발견하고, 사회적인 욕구(needs)를 새로운 시장으로 재인식함으로써 경제적 가치와 사회적 가치를 창출하는 일련의 과정을 거쳐, 사회적 문제를 해결함과 동시에 사회적 이슈에 관한 관심으로 이어질 수 있는 윤희제의 역할을 하는 가상 기업을 설립하며 조직변화 전반에 대한 이해를 주도적으로 도모할 수 있었다. 결국 조직변화 수업으로, 트렌드에 대한 인사이트를 발견하여 문제를 다면적으로 바라보고 수업 내용과 결부하여 해결방안을 찾아내고, 내면화할 수 있었다.

더불어 기업과 결부된 경영학적인 관점에서 나아가, 개인적 측면에서의 CSV에 대입하여 스스로의 가치 체계를 정립하고 적용해 볼 수 있었다. 즉 ‘나의 미션, 비전, 핵심 가치는 무엇인가?’ 라는 질문에서 파생되어 ‘나는 나를 어떻게 경영할 것인가?’ 에 대한 고찰로 나아갈 수 있게 하였다. 이전까지 나는 개인의 삶이, 연속적인 것이 아니라, 그저 순간순간에 존재할 뿐이며, 어쩌면 이는 기업적인 측면에서 바라보아도 마찬가지일 것이라는 생각을 해왔다. 그러나 순간순간이 모여 ‘삶’ 전체를 일관한다면, 기업의 모든 결정의 기준이 되는 가치 체계를 설립하는 것과 같은 맥락으로 개인의 삶의 가치관과 가치 체계를 정립하는 것의 중요성을 체감할 수 있었다. 나의 가치관이 나의 판단의 기준점이 되어 결국에는 앞으로 있을 나의 모든 결정에 관여할 것이고, 이러한 결정들이 모여 결국에는 ‘삶’ 이라는 연속체를 이룰 것이다. 그러나 기업과 달리 인간의 존재 이유, 존재의 본질을 규정하는 것은 쉬운 일이 아니다. 기업은 목적을 가지고 태어났으나, 우리가 태어난 목적, 우리의 존재 이유는 어쩌면 영원할 허상의 영역이다. 실재하더라도 끝내 무지의 영역으로 존속될 것이다. 이에 기업의 지속성과 마찬가지로 나의 지속성을 위해 본질적인 목적과 더불어 추구해야 할 가치는 무엇인지 묻게 된다. 즉, ‘나를 지속 가능하게 해주는 가치는 무엇인가?’ 라는 물음이다. 결국, 나의 존재의 본질을 규정하는 것은 불가능에 가까울지 몰라도 ‘내가 세상에 존재함으로써 어떠한 영향력을 미치고 싶은가?’, ‘나는 어떠한 가치관을 사유하고 어떤 가치관에 따라 살아가고 싶은가?’ 에 대해 스스로 묻고 답하는 과정을 통하여 나만의 가치 체계를 설립함으로써, 주체적인 삶으로서의 나를 경영하게 되고, 가치관에서 파생된 결정, 습관들이 삶을 일관하게 되며 결국에는 나 자체가 될 것이라는 생각에 닿게 되었다. 삶에 대한 관조

적 태도는 결국 내 인생의 주인, 내 인생을 경영하는 데에 있어서 스스로가 CEO가 될 수 없다는 것을 의미하게 될 것이라는 깨달음을 얻게 되었다. 즉 주체적으로 삶을 경영함으로써 성공의 데이터, 실패의 데이터를 축적하여 외부의 영향에 의해 흔들리지 않는 ‘자기 확신’의 기반을 마련하는 것이 나에게 ‘경쟁력’이 되어 개인으로 하여금 지속 가능한 가치를 추구할 수 있도록 할 것이라는 생각에 닿게 되었다. 나아가 기업의 가치 체계에서 기업의 단기적인 목표(VISION)의 상위에 존재 이유(MISSION)를 규명하는 것으로 보아, 개인의 단기적인 직업, 꿈이 목표가 되는 것이 아니라, 내가 이루고 싶은 ‘가치’가 목표가 되어야 할 것이라는 사실을 깨닫고 사유할 수 있었다.


‘조직변화’ 과목의 원격수업으로 얻은 경영학에 대한 통찰과 평소 관심 분야인 예술 경영학문과의 결부로 ‘나를 어떻게 경영할 것인가?’에 대한 고민의 기회를 마련할 수 있었던 기회는 삶에 큰 자양분이 될 것이다.

2. 나만의 학습 노하우


시험과 중간 평가 모두 보고서 제출 형태로만 이루어졌던 본 수업에서는 평가 방식에 ‘정답’이 없다는 걸 염두에 두고 강의내용을 바탕으로 많은 자료를 찾아서 참고하고, 의견이나 생각, 경험을 중심으로 독창적으로 내용을 전개하는 것이 중요하다고 판단하였다. 형식상으로는 주어진 양식이 없더라도, 다음과 같이 제목과 과목명, 담당 교수, 소속, 학번, 이름, 제출일, 페이지 표지가 포함된 표지를 만들고, 신뢰를 줄 수 있는 글꼴을 사용하여 보고서에 좋은 인상을 줄 수 있도록 하여 보고서 양식을 제작하였다. 또한 표지의 다음 페이지에 목차를 작성하여, 글을 작성하고, 평가함에 있어 전체적인 흐름을 파악할 수 있도록 하는 작업을 통해 보고서 내용을 전개하는 데에 있어 많은 도움이 되었다.

조직변화(OC)-기말고사 대체과제

[공유가치창출(CSV)에 관한 종합적 고찰]



과목명	조직변화(1076-1)
담당교수	이태진 교수님
소속	E-business학과
학번	201921532
이름	신선아
제출일	2021년 6월 20일




목차

- 1 서론: 초록
- 2 본론: 공유가치창출(CSV)의 의미와, 공유가치창출(CSV)전략 수립방법
- 3 결론: '조직변화(Organizational Change)'과목의 공유가치창출(CSV)관리를 수강하고 얻은 도움

1 (류성만,권장숙, 2009; 조윤형,최우재, 2011; 채신석, 2014; Ba
O'Connell, Hickerson, & Pillutla, 2011; Slack, Orlic, & Anders

2
3
<https://namu.wiki/w/%EB%A8%9C%9C%AD%EB%A7%A4%20%EC%82%AC%EA%B1%B4>



3

보고서 작성에 특히 신경 쓴 부분은 서론부의 ‘초록’과 결론부이다. 국문 초록 부분은 본고의 본론부, 결론부에 전개될 내용과 얻은 시사점에 대하여 간단하게 밝힘으로써 읽는 이(평가자)로 하여금 수월한 이해를 도울 수 있다. 또한 내용적으로, 결론부에 깨달음과 시사점을 전개하며, 간 학문적인 탐구를 도모하기 위한 노력을 지속하였다. 타 수업인 ‘조직문화’ 수업에서 배운 이론을 다음과 같이 적용시키고, 첨부하여 간 학문적인 통찰을 얻을 수 있었다.

더불어 결론부에 실제 사례와, 실제 사진을 첨부하여 본문의 이해를 도운 점도 좋은 평가를 받을 수 있었던 요소라고 여겨진다.

<p>사용하기로 하였다. 나이가 많았을 때 번번하게 사용되기 위해 화장대의 질이 낮아 너무 색칠과 비후한 소재는 필요를 찾아내어 방석으로 만들어 사용하였다. 더불어 기존 화장대의 스탠드로 조명의 역할을 놓치지 않으면서 그림 6, 7, 8의 최후적인 화장대 모습이 완성되었다.</p> <p>이러한 원천의 과정을, 직식화장대를 입식화장대로 만드는 과정들을 거치며 기존의 것들을 새롭게 결합하여 유용성을 만들어 낸 것이다.</p>  <p>그림 5- 화장대 스톤</p>  <p>그림 6, 7, 8- 입식 화장대</p>	<p>2. 본론</p> <p>‘직식 화장대를 입식 화장대로 바꾸려면 어떻게 해야 할까?’ 나는 이러한 질문을 해결하는 과정, 즉 직식 화장대를 입식 화장대로 만드는 과정에서 나의 ‘행의’를 발견해 낸 경험이 있다.</p> <p>평소 나를 둘러싼 공간이 가진 힘과, 그러한 공간을 나의 취향 및 꾸미는 인테리어 작업에 관심이 많았던 나는 직업을 기점으로 10명 남짓한 북촌 오 피스텔에서 생애 첫 독립을 하게 되었다. 자취방의 인테리어를 구성하는 과정에서, 원래 살고 있던 본가에 있는 모든 짐을 전부 들여놓기에는 공간이 부족했기에, 그림 1에서와 같이 공간 활용도를 높이기 위해 방 중앙에 공부와 식사를 겸하는 큰 크기의 다이닝 테이블을 놓아 식탁으로도, 책상으로도 사용할 수 있게 하여 공간 활용도를 높이기도 하였다. 그러나 공간을 구상하며 분할한 문재점은 평소 의출 준비에 많은 시간을 할애하는 화장대를 놓을 공간이 부족했던 것이었다. 평소 본가에서 사용하는 큰 화장대를 가져다 놓기에는 공간이 협소하게 느껴졌고, 다이닝 테이블에 화장대의 용도까지 추가하기에</p>  <p>그림1- 다이닝 테이블</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

또한 시사점이나 느낀 점을 다룰 때에 있어 전공, 관심 분야와 관련된 최근의 트렌드와의 결부를 통해 ‘유용성’이라는 가치로 과목 내용을 내재화하기 위하여 노력했다. 예컨대 소비자의 형태가 가치 소비로 변화하면서, CSV는 기업과 소비자를 동일시하여 브랜드 이미지에 영향을 준다는 관계를 밝혀낸 연구를 바탕으로, 최근 COVID-19로 인한 공유가치창출(CSV)에 관한 시사를 통해 기업 차원에서의 CSV뿐만 아니라 소비자의 관점에서 공유가치창출(CSV) 개념을 이해하였다. 또한 Covid-19로 인해 각종 산업 업종들의 직접적인 타격을 받으면서, 경제 위기와 동시에 소비 심리가 위축되고, 고용 환경이 악화되는 등의 악영향이 장기적으로 지속되고 있다는 점에 주목했다. Covid-19로 인한 정부 권고 수칙인 사회적 거리두기로 인해 발생한 언택트(Untact)와 홈코노미(Home+Economy) 현상으로, 코로나19 이전에 증가하고 있던 온라인 쇼핑 트렌드가 가속화되어, 백화점과 대형마트 등 오프라인 플랫폼을 기반으로 한 구매는 급락한 반면, 온라인 플랫폼 기반 구매는 급증하고 있음을 타 과목에서 분석했던 경험을 떠올릴 수 있었다. 이에 covid-19로, 온라인 채널 위주의 소비패턴이 더욱 명확해지면서, 기존의 오프라인 플랫폼이 가진 경쟁력과 공동가치 창출의 비즈니스의 가치를 온라인 플랫폼으로 이동시키는 것이 주요한 과제로 대두하였음에 주목하였다. 따라서 기존의 고객 데이터를 활용하여 개인화되고 차별화된 경험을 제공하는 통합적인 접근 전략의 필요성을 시사하였다. 또한, 기업은 디지털 고객경험(Digital CX)에 대한 정밀 분석을 토대로 각 고객의 개인화된 니즈를 파악하여 소비자 관점의 커뮤니케이션에서 어떠한 고객 경험을 제공할 것인지에 대한 로드맵을 수립해야 할 것임을 고찰할 수 있었다. 나아가 비즈니스는 소비 트렌드 변화를 파악하여 지속 가능한 성장을 위하여 CSV 전략뿐만 아니라 디지털 고객 경험 분석, 데이터 분석 등을 기반으로 하는 전략을 수립함으로써 소비자 관련 다각적 데이터를 수집, 통합, 분석하여 심층적으로 이해하는 비즈니스를 전개하여야 할 것임을 통찰할 수 있었다. 이와 같이 변화하고 있는 소비자의 행동 변화 양식에 대하여 고찰하고, 가속화되고 있는 포스트 코로나 산업 트렌드를 파악하여

비즈니스 모델과 시장 진출 전략의 전망에 대한 시사를 얻기 위한 노력을 기울였으며, 종합적으로 조직변화 수업에서 학습한 지식을 간 학문적으로 확장함과 동시에, 재해석을 통해 조직변화 수업의 내용을 유용성이라는 가치로 내재화함에 의의가 있었다.

끝으로, 아주인들로 하여금 포스트 코로나 시대의 도래로 인해 변모한 언택트(Untact) 수업 방식에 적응하기 위해서는, 스스로가 강의의 의식적인 주체가 되어야 할 것이라는 깨달음을 전하고 싶다. 『자유로부터의 도피』에서 에리히 프롬은, 어딘가에 속해 있지 않고 삶이 어떤 의미와 방향도 갖지 않으면 자신의 존재가 한낱 티끌처럼 느껴지고 무의미하다는 느낌에 압도당하게 된다고 말했다. 원격수업으로 인한 의사소통의 한계와 소속감의 결여, ‘강의실’이라는 실체의 부재 아래에서, 학생으로써 ‘주어짐’의 객체가 된다는 것은, 또 그러한 기분에 사로잡히게 된다는 것은 그리 이상한 일이 아니다. 따라서 우리는 수업에서 학습한 지식들을 간학문적으로 확장하고 그에 대한 인사이트를 발견하여, 수업의 의미와 방향을 설정하는 ‘의식적인 주체’가 되어야만 ‘언택트(Untact)’ 시대에서 ‘온택트(Online Conact)’로 변모하는 포스트 코로나 시대의 흐름을 파악하며 ‘온택트(Ontact)’ 시대의 장점을 잘 활용할 수 있게 될 것이다.

3등

문석 교수님과 뽀모들

교과목명 : 고분자과학

교수명 : 김문석

학과 : 응용화학생명공학과

이름 : 최유빈

1. 수업 참여 후기

김문석 교수님의 고분자 과학 과목을 수강하며 교수님과 여러 교감을 나누었다고 생각한다. 우선, 교수님께서 학생 모두 카메라를 켜길 권하신다. 학생을 이름으로만 보는 것이 아닌, 얼굴과 이름을 함께 매치시키며 수업을 진행하시면서 이름을 불러주신다. “유빈 학생이 질문한 내용인데요.”, “유빈 학생, 화면 잘 보이나요?” 와 같이 수업 중간중간 이름이 불리면 정신도 번쩍 차려지고, 내가 집중하는 모습이 교수님께 그대로 보이는 것이 느껴져 더 수업에 집중하게 된다. 그리고 강의 중 기지개 펴기, 물먹기 권유를 많이 하신다. 예를 들면 “우리 잠깐 기지개 좀 펴고 갈까요?”, “물먹고 합시다.” 처럼 말씀을 하시면 수강생들이 모두 기지개를 피우거나 물을 먹는데, 먹지 않는 사람이 있으면 “유빈 학생, 물먹으면 좋아요.” 와 같이 말씀을 해주신다. 그래서 ‘아 교수님은 학생들에게 고르게 신경을 쓰시는구나.’ 라고 생각하게 된다. 물이나 기지개를 피우는 타이밍은 주로 조는 학생들이 많거나, 어려운 내용을 시작하기 전, 어려운 내용을 설명하시는 중간에 주변을 환기 시키기 위해서 일 때가 많다. 그래서 졸다가 기지개를 피우는 시간이 되면 ‘아 역시 나만 집중이 안 되는 게 아니구나.’, ‘나만 어려운 게 아니구나.’ 라는 생각을 하면서 다른 학생들과 멀리 떨어져 있지만 같은 강의실에 있는 것처럼 동질감이 든다. 수업을 듣고 메일로 질문을 보내드리면 보낸 메일에 정성스레 답변을 해주신다. 또한 내가 보낸 메일 속 질문 밑에 바로 답변을 보내주시는 형식으로 답변해주셔서, 어떤 질문에 대한 답변인지, 어느 포인트에서 내 질문이 잘못됐는지 쉽게 알 수 있다.

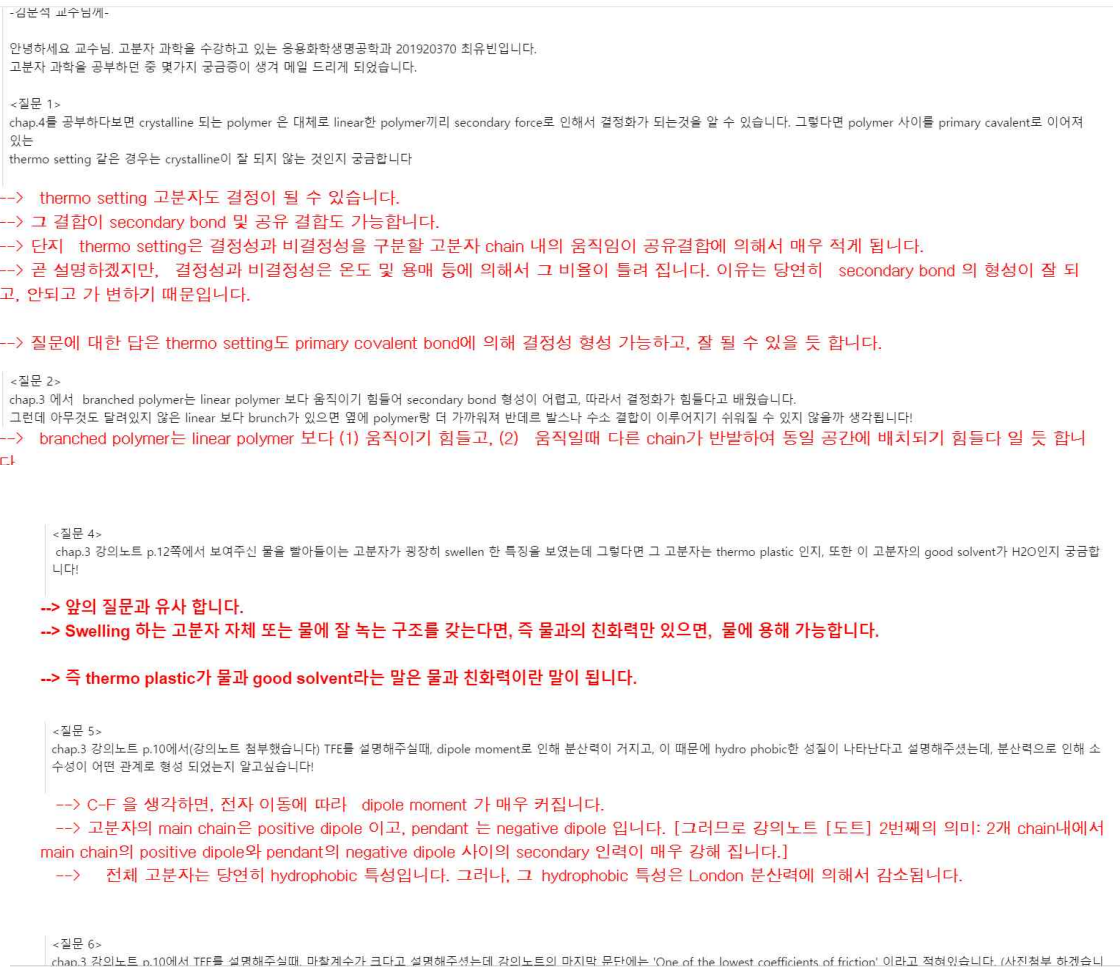


사진 1. <김문석 교수님과 질의응답 메일 중 일부>

위의 사진은 김문석 교수님께 수업 내용 중 궁금한 것과 수업 외의 궁금한 점을 여쭙본 메일로, 검은색 텍스트는 질문, 빨간색 텍스트는 질문에 대한 교수님의 답변이다. 질문은 6개로 한꺼번에 많은 질문을 드렸음에도 불구하고, 모든 질문마다 자세하고 상세하게 설명해주셨다. 또한 긴 질문들 밑에 빨간색으로 교수님의 답변을 달아주셔서 답변을 읽기 편하고, 내가 쓴 질문과 비교하며 답변을 확인할 수 있어 매우 효과적인 답변 방식이라고 생각했다. 질문을 드리고 답변을 받은 후에도 질의응답은 끝나지 않고 수업 중에서도 계속된다. 앞서 질문했던 관련된 내용이 수업에 나오면 “유빈 양이 질문했던 질문이죠? 유빈 양은 A라고 생각했지만 B라고 생각해서 C라는 성질이 나올 수 있어요.” 라고 질문한 사람을 호명하시며 질문에 대해 간략하게 설명해주시고, 질문에 대한 답변과 수업 내용을 다시 엮어서 설명해 주신다. 질문한 사람은 자신의 질문이 교수님으로 인해 언급되면서 복습과 질문의 중요성을 알 수 있고, 다른 학생들은 다른 사람이 한 질문을 들으면서 지식을 확장시킬 수 있다. 나는 수업시간에 배운 내용을 바탕으로 혼자 생각해 보고 일상생활에 대입

하는 걸 좋아한다. 예를 들면 고분자에서 배운 poly ethylene을 뼈빠로 포장 용기에서 발견하거나, poly carbonate를 투명 하드케이스에서 발견하며 투명하고 내구성이 좋은 특징으로 인해 케이스에 쓰이는 것을 다시 떠올리는 것을 즐긴다. 이렇게 수업시간에서 배운 내용 외의 질문이 생기는 경우를 교수님께 여쭙보아도 정성스럽게 답변해주시는 문석 교수님의 답장을 받으면 더 열심히 공부해서 좋은 질문을 드리고 싶다는 생각이 든다. 마이크 샌델 강의에서는 질문한 학생 이름을 강의 중 불러주며 질문자를 주인공으로 만들어 학생들에게 질문의 중요성을 알려준다고 한다. 문석 교수님 또한 질문자를 주인공으로 만들어줌과 동시에, 질문의 내용과 답변을 공유하시며 학생들의 지식을 확장시켜 주신다. 또한 학생들이 기지개와 물을 먹는 행동을 같이 하게끔 이끌어 주시면서 함께 수업을 듣고 있는 동질감과 전우애를 심어주시는 강의라 생각한다.

2. 나만의 학습 노하우

학습 노하우 1) 교수님께 질문하기

:많은 학생들이 교수님께 질문드리기를 무서워하고, 메일 드리는 것을 꺼려 하지만 어렵지 않다. 이해가 느린 편인 나는 수업 중, 공부 중 생긴 궁금증은 고민 후 교수님께 여쭙보아 해결하는 편이다.

전공 수업은 모든 단원을 다 깊이 있게 설명하기엔 범위가 넓고 단원 수가 많기 때문에 관심이 있거나 취업과 관련 있는 분야는 스스로 여쭙보아야 한다. 강의를 듣기만 하는 것보다는 강의내용을 복습하고 이해가 되지 않는 부분을 여쭙보면, 내가 알고 있던 것보다 훨씬 깊은 전공 내용이 숨겨져 있다는 걸 알게 된다. 또한 A 전공과 B 전공 사이에 관계가 있다는 것을 깨닫고, 한 과목만을 공부하는 것이 아닌 다른 과목과 연결하여 서로의 전공지식을 대입할 수 있다. 이렇게 서로 다른 전공지식들을 연결하면서 공부하면 점차 이해되는 폭이 넓어지고, 질문하기 전보다 넓은 시각으로 과목을 바라보고 공부할 수 있다.

교수님께 메일을 보내는 것이 두렵고 어려울 수 있지만 기본적인 예의만 지키면 어려운 일이 아니다. 지켜야 할 사항들은 다음과 같다.

<메일 쓰는 법>

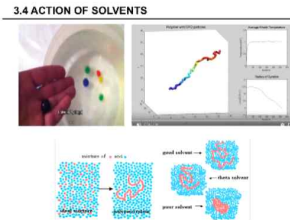
1. 제목은 요점을 담아 알기 쉽게 작성 예) [고분자과학] 수업 질문
2. 받는 분에 대한 예의를 지키기 예) -김OO 교수님께-
3. 자신이 누구인지 설명하기 예) 고분자 과학을 수강하는 (학과) (이름) (학번)

4. 메일을 보낸 이유를 명확히 하기 예) 7/4 강의 중 ~~ 부분을 질문드립니다.
5. 마지막으로 예의 지키기 예) -최O빈 올림-

위와 같은 기본을 지켜 메일을 쓴다면 받으시는 교수님도, 드리는 학생도 모두 예의를 지키는 질의응답을 할 수 있다.

이외에도 비대면 학습으로 교수님께 직접 교재를 보여드릴 수 없기 때문에, 쉽게 질문의 요지를 파악하실 수 있도록 메일에 내가 궁금한 부분의 강의노트를 첨부하는 것도 좋은 팁이다.

<질문 3>
chap.3에서 보여주신 중영상에서 물을 빨아들이는 고분자 중영상을 보여주셨습니다. (강의노트 p.12 밑에 첨부하셨습니다) 이때 물을 빨아들이는 고분자를 만들고 싶다면 이온화된 monomer를 쓰는것이 적합하다고 말씀해주셨는데, 그 이유가 이온화된 monomer를 사용해 고분자를 형성하면 물을 빨아들여 monomer끼리의 거리가 멀어져도 이온화 된 monomer (secondary bond-ionic bond)가 잘 유지되어 원래 형태를 유지 하기 때문에



이온화 된 monomer 가 적합한 것이 맞는지 궁금합니다. 혹시 틀렸다면 이유를 알려주시면 정말 감사하겠습니다 :)

<질문 6>
chap.3 강의노트 p.10에서 TFE를 설명해주실때, 마찰계수가 크다고 설명해주셨는데 강의노트의 마지막 문단에는 'One of the lowest coefficients of friction' 이라고 적혀있습니다. (사진첨부 하였습니다) 어떤 것이 맞는것인지 알려주시면 감사하겠습니다 :)

- > 미안해요. 수업중에 표현을 잘 못 했나 보군요.
- > 강의노트처럼 [적다] 가 맞습니다.

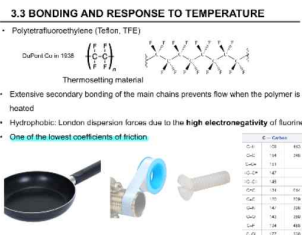


사진2. <고분자 과학 질의응답 메일-강의노트 첨부>

위의 사진은 고분자 과학을 공부하던 중 이해가 되지 않은 강의노트를 캡처하여 메일에 함께 보낸 내용 중 일부이다. 질문을 할 때 강의노트의 페이지를 언급하고, 강의노트를 사진으로 첨부해 교수님께서 쉽게 질문의 요지를 파악하실 수 있도록 노력했다. 교수님의 수고를 덜어드릴 수 있도록 배려한다면 서로 배려하는 메일을 통한 상호작용을 이룰 수 있을 것이다.

학습 노하우2) 동기들과 공부하기 - 뽀모도로 공부법 (pomodoro)

:코로나 팬데믹으로 인해 가장 걱정되었던 부분은 옆에서 함께 공부하는 동기들이 없어 승부욕이나 시험 기간에 대한 현실감이 떨어지는 것이었다. 항상 나를 바로잡기 여러 글귀나 명언을 책상 앞에 붙였지만, 의욕이 떨어지면 공부하고자 하는 의지 또한 사라지고 이는 곧 나에 대한 죄책감으로 자리 잡게 되었다. 그래서 동기들과 함께 공부하기로 마음먹고, pomodoro(뽀모도로) 공부법을 함께하기로 했다. 뽀모도로는 25분 동안 집중하고, 5분 동안 쉬는 공부 방법이지만 우리는 50분 동안 공부하고 10분 동안 쉬는 뽀모도로를 했다. 동기들 4명이 모여 zoom을 열고, 유튜브에서 pomodoro를 검색한 다음 화면공유를 통해 타이머를 켜다.

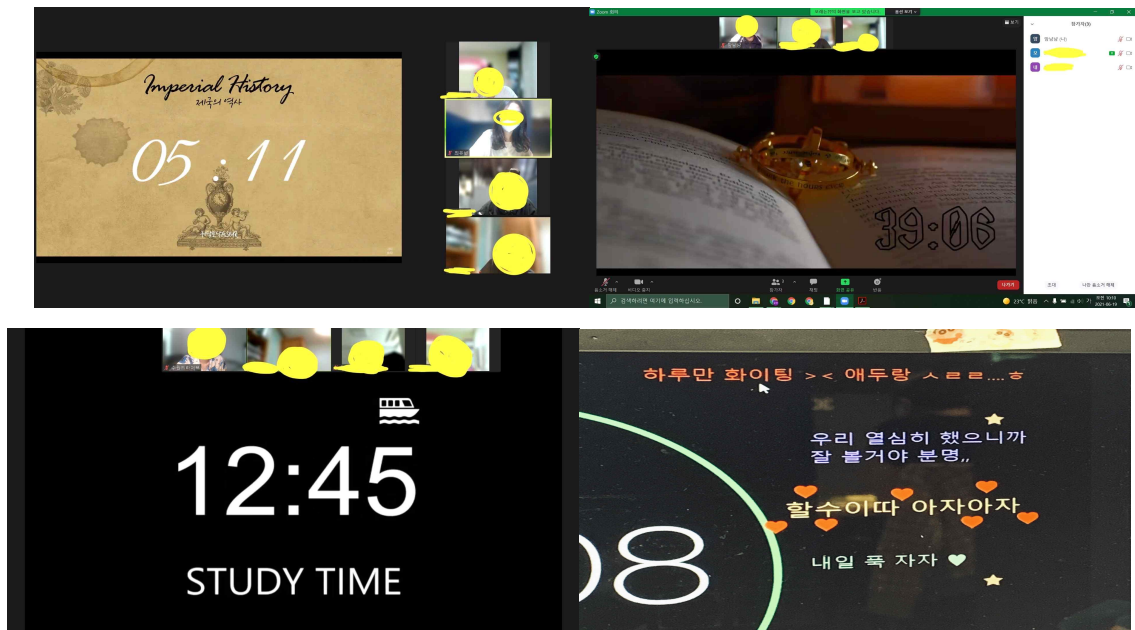


사진3. <뽀모도로를 하는 zoom화면>

한 학기 동안 시험 기간 한 달 전부터 매일 zoom을 켜고 유튜브에서 뽀모도로 화면공유를 했다. 동기들이랑 단체 카톡방을 만들어 서로를 깨우고, 수업이 없는 주말이면 오전 8시부터, 수업이 있는 날이면 수업 끝나고 바로 zoom을 켜 모두 공부하는 환경을 조성하기 위해 노력했다. 동기가 졸고 있으면 깨워주고, 너무 피곤한 동기는 휴식 후 일어나라고 연락하며 서로를 이끌어갔다. 함께 듣는 전공이 많아 서로 모르는 걸 질문하고 답해주기도 하고, 교수님께 답변을 받으면 나누면서 전공 지식을 쌓기 위해 힘을 모았다. 노력에 보상받지 못하거나 유독 힘든 날들은 동기들이 집중해서 공부하는 모습을 보며 스스로를 다독였다. 또한 어려운 공부를 위해 노력하는 동기들의 모습을 보며 전공 수업이 나에게만 어렵고 힘든 것이 아닌, 모

든 사람들에게 어려운 것이 일반적이므로 노력하면 할 수 있음을 계속해서 되뇌 수 있었다.

50분간 열심히 공부하고, 10분 동안 쉬거나 공부하던 중 모르는 것이 나왔을 때 토크방에 질문을 하였다. 공부하던 중 모르는 것이 나왔을 때 단체 카톡방에 질문을 올리면 10분 동안 토론의 장이 열린다. 질문에 대한 답변을 나누며 내가 공부했던 것을 복습하거나 몰랐던 것을 알게 되어 오개념을 바로잡을 수 있는 기회가 된다. 그 후 다시 깨달음을 얻은 상태로 50분 동안 집중해서 공부하면 효율은 더 오르게 된다.

서로 알려주고 공부하는 것 외에도 시험 기간을 함께 보낸다는 동지애가 생겨 힘든 기간을 더 잘 버틸 수 있게 해준다. ‘하루 더 한다고 달라질까?’, ‘너무 떨려. 어떡하지?’ 와 같은 생각이 들 때, ‘하루만 더 힘내자.’, ‘우리 열심히 했으니까 잘할 거야.’ 와 같은 동기들의 말을 들으면 ‘그래. 여기까지 달려온 거 좀만 더하자, 난 잘할 수 있을 거야.’ 라는 다짐을 하며 마지막의 마지막까지 최선을 다하게 된다.

코로나로 인해 동기들과 캠퍼스 추억을 쌓지는 못했지만, 아침마다 서로 인사하며 zoom을 켜고, 즐기고 있는 서로의 모습을 놀리고, 놀고 있으면 얼른 공부하라며 잔소리해주는 동기들과 함께한 한 학기는 내게 많은 추억을 쌓게 해주었다. 이러한 원동력이 있었기에 어려운 전공 수업들과 많은 과제량을 함께 버틸 수 있었다고 생각한다. 따라서 비대면 공부 방법 노하우 중 가장 추천하고 싶은 방법은 친구들과 뽀모도로를 하는 것이라 자신 있게 말할 수 있다. 마지막으로 한 학기 동안 정말 열심히 달려온 현주, 소연, 수연에게 정말 고마웠고, 다음 학기도 잘 부탁한다는 말을 전하며 비대면 학습 노하우를 마친다.

3등

다른 어느 과목보다 열정을 기울이고 최선을 다했던 객체지향프로그래밍 및 실습수업

교과목명 : 객체지향프로그래밍및실습

교 수 명 : 정찬기

학 과 : 국방디지털융합학과

이 름 : 하정민

1. 수업 참여 후기

입학과 동시에 코로나가 발생하여 겨울방학 계절학기 수업 외에는 학교에서 대면으로 강의를 들은 적이 거의 없습니다. 비대면으로 동영상 강의를 듣는 것이 익숙해졌다고 하지만 아직도 어색한 것이 사실입니다. 이렇게 비대면으로 수업이 전환되면서 수업 방식은 크게 3가지로 분류되었습니다. 그중에서 제가 수강한 객체지향프로그래밍 및 실습은 1주일에 3번 실시간으로 수업이 진행되었습니다. 1학년 때는 대학 공부와 학과 필수 공부에 기반이 되는 과목들을 수강하였기에 비교적 이해하기 쉽고 고등학교에서 배운 내용과 연관되어 있기에 친숙하게 다가왔습니다. 하지만 2학년 1학기에 배운 객체지향프로그래밍 및 실습에서는 처음으로 JAVA를 배웠기에 많은 어려움을 겪었습니다. 수업은 월요일, 수요일, 금요일 총 3번씩 진행되었으며, 월요일에는 실습을 하였고 수요일과 금요일에는 이론 수업을 들었습니다. 이 수업을 들으면서 가장 좋았던 점은 학생과 교수님 사이의 소통이 원활하고 학생이 이해한 정도를 교수님이 실시간으로 확인하면서 수업을 진행하기에 공부를 하기에 수월했다는 점입니다. 수요일과 금요일에 수업한 이론을 바탕으로 월요일에 이와 관련된 JAVA 프로그래밍 실습을 하기에 이해하기에 어려움이 없고, 이론 수업이 끝나기 전마다 교수님께서 Q&A 시간을 가져 이론에 대한 궁금증과 어려움을 해결해주셨습니다. 그리고 월요일에 하는 실습 시간에는 예제를 풀 때, 학생 모두가 예제를 학습했는지 확인해주시며, 어려운 부분이 있을 때는 학생이 화면공유를 하여 문제점과 어려움을 설명하고 이에 대한 피드백을 주는 시간을 가졌습니다.

저는 이번 학기에 들은 7개의 과목 중에서 객체지향프로그래밍 및 실습이 가장 학생과 교수님 사이의 상호작용이 원활했습니다. 실제 저도 월요일 실습 시간에 어

려움을 겪었던 점이 있습니다. 아직도 기억에 나는 부분인데 자동차와 속도 등에 대한 코드를 프로그래밍할 때 speed가 60이 나와야 하는데 0이 나와 계속 노력을 했지만 코드 결과가 바뀌지 않아 어려움을 겪고 있었습니다. 그때 어려움을 겪는 학생은 채팅으로 어느 부분이 어려운지 알려달라는 말씀을 교수님이 하시자 제가 바로 채팅으로 speed 값이 올바르게 나오지 않는다고 말하였고 이를 화면공유를 통하여 보여드렸습니다. 그래서 저의 화면을 유심히 확인하시고 문제점이 그 위에 있는 생성자(클래스라는 자바 파일로부터 객체를 생성할 때 호출되어 객체의 초기화를 담당하는 역할)에 있다고 설명해주시고 제가 문제점을 해결할 때까지 기다려주셔서 원활하게 수업에 참여할 수 있었습니다. 이때 정말 큰 감동을 받았고 비대면 수업이었지만 화면공유와 학생과 교수님 사이의 소통을 통해 어려움 없이 수업을 들을 수 있다는 것을 깨달았습니다. 그리고 이 외에도 첫 수업시간에 JAVA를 실행하기 위하여 이클립스라는 프로그램을 설치하였던 경험이 있습니다. 이때 환경변수와 경로 설정으로 인해 많은 어려움을 겪었는데 교수님께서 수업시간이 이미 지났음에도 불구하고 어려움을 겪을 학생들을 모아 한 명 한 명 문제점을 확인해 이클립스를 완벽히 설치할 수 있도록 도움을 주셨습니다. 이처럼 매 수업시간마다 제가 부족한 부분이 있더라도 이를 채워주시고 이론에 대한 궁금증과 어려움을 해결해주셔서 기억에 남는 수업이었습니다.

또한 객체지향프로그래밍 및 실습수업은 총 3번의 시험과 3번의 큰 과제가 있었습니다. 이렇게 많은 시험과 과제가 한 학기 동안 진행되었는데, 이로 인하여 JAVA를 매일매일 공부를 하여야 했습니다. 많은 시험과 과제로 인하여 한 학기 동안 JAVA가 머리에서 지워지지 않고 매일 머리에 남아있었는데 이는 교수님의 큰 뜻이었다고 생각하고 있습니다. 이처럼 시험과 과제가 끝난 다음 날에는 꼭 시험과 과제에 대한 피드백을 교수님께서 해주셨습니다. 시험과 같은 경우에는 1번부터 마지막 문제까지 하나하나 설명을 해주시면서 저희들이 자주 하는 실수 등을 알려주셨고 이 과정에서 질문을 하면 피드백까지 해주셔서 시험에 대한 궁금증을 풀 수 있었습니다. 그리고 과제와 같은 경우에는 한 과제당 거의 일주일의 넘는 시간이 소요되는 큰 과제들이었는데 이 과제들이 끝난 후에도 과제에 대한 전반적인 설명과 질문에 대한 피드백을 해주셨습니다. 저는 이렇게 많은 시험과 과제 중에서 마지막 과제였던 쇼핑물 만들기 프로젝트를 진행할 때 가장 큰 도움을 받았습니다. 이 과제는 마지막 과제였기에 여태까지 배운 모든 JAVA 지식들이 동원되는 과제였습니다. 그렇기에 많은 시간과 지식이 필요했습니다. 그중에서도 저는 GUI라는 부분에서 어려움을 겪었습니다. 여기서 GUI란 그래픽 사용자 인터페이스라는 뜻으로 사용자가 편리하게 사용할 수 있도록 입출력 등의 기능을 알기 쉬운 아이콘 따위의 그래픽으로 나타낸 것으로 이번 쇼핑물 프로젝트에서 웹사이트적인 측면을 담당하는 기능을 뜻합니다. 이 GUI는 수업시간에 이론적으로 배우긴 했으나 그 범위가 광범

위하여 모든 기능을 익히기에는 어려움이 있었습니다. 하지만 제가 할 수 있는 한 최대한의 노력을 기울여 과제를 마무리하였는데 과제를 끝낸 후 수업에서 교수님께서 이 과제에서 주목해야 할 점과 진행하면서 필요한 지식들을 정리해 설명해주셨기에 과제에 대한 궁금증이 풀리게 되었습니다. 이처럼 한 학기 동안 JAVA에 대한 모든 내용을 배웠다고 말할 수 있을 만큼 교수님께서 객체지향프로그래밍 및 실습 수업을 최선을 다하여 수업해주셨고 학생들에게 끊임없이 피드백과 상호작용을 해주셔서 정말 좋은 경험이 되는 수업이었습니다. 감사합니다.

2. 나만의 학습 노하우

객체지향프로그래밍 및 실습수업은 4학점이기에 일주일에 총 3번의 수업이 있습니다. 3번의 수업 중 2번은 이론, 1번은 실습수업이기에 각 수업별 연습 및 복습 방법을 먼저 구축하였습니다. 이론 수업과 같은 경우에는 교수님께서 미리 강의노트를 올려주시기에 연습이 가능합니다. 이 강의노트 대부분이 영어로 되어 있기에 저는 수업이 있는 날 아침에 일어나자마자 바로 강의노트를 공부하였습니다. 강의노트를 프린트를 통해 뽑거나 태블릿을 이용하는 등 다양한 방법으로 공부를 시작해 보았는데 그중에서도 저는 태블릿을 통해 강의노트를 읽어보는 방법이 가장 좋았습니다. 수업이 시작하기 전까지 오늘 배울 내용이 무엇인지 미리 읽어보고 강의노트가 영어로 되어 있기에 이를 대비하여 어려운 부분은 해석하여 기록해놓았습니다. 그래서 이를 기반으로 이론 수업을 들었습니다. 그리고 이론 수업을 들으면서는 교수님의 말씀 및 교수님의 필기를 받아적었고 수업이 끝난 후에는 제가 스스로 내용을 정리해보는 시간을 가졌습니다. 밑에 사진들을 보시면 알 수 있듯이 저는 저만의 노트를 제작하는 것이 공부하는데 큰 도움이 된다고 생각하여 저는 항상 공부를 할 때 제 노트를 따로 작성합니다. 이렇게 정리를 하면서 이론 수업을 복습을 하며, 실습 수업과 같은 경우에도 먼저 강의노트를 미리 교수님께서 올려주시기에 이를 바탕으로 오늘 배울 내용을 학습하였습니다. 그리고 중간 중간에 연습 프로그램이 있기에 이를 스스로 해석해보고 어려운 부분이 있으면 찾아보면서 수업 전에 수업 내용을 이해하고 들어갔습니다. 수업시간에는 교수님의 말씀을 들으면서 JAVA 프로그램을 다시 한번 풀어보고 내가 찾아본 내용과 일치하는지 비교하면서 학습하였습니다. 그리고 수업 후에는 코드를 직접 노트에 옮겨 보기 쉽게 정리하였습니다.

이처럼 연습과 복습, 수업시간 경청을 하면서 객체지향프로그래밍 및 실습수업에 빠져들게 되었습니다. 제가 이 과목을 공부할 때 가장 중요하게 생각했던 것은 바로 저만의 노트를 제작하는 것이었습니다. 밑에 있는 자료들을 보시면 알 수 있는데 이론 수업 노트와 실습 수업 노트를 구분하여 정리했습니다. 이론 수업과 같은 경우에는 저는 개념 위주로 정리했습니다. 이론 수업을 교수님께서 하실 때 개념과

더불어 이에 대한 응용을 강조하셨기에 저는 이론을 완벽히 알려면 강의노트와 수업에서 배웠던 개념이 기반이 되어야 한다고 생각했습니다. 그래서 용어 정리 및 JAVA에서 사용되는 기능들의 역할 등을 하나하나 빠지지 않고 기록했으며, 그림 자료가 필요한 경우에는 직접 찾거나 강의노트에서 가져와 옮겼습니다. 이렇게 개념을 탄탄하게 한 결과, 시험에서 나오는 개념 관련 문제를 수월하게 풀 수 있었으며, 수업 자체가 실습을 진행하는 수업이기에 JAVA를 통해 직접 프로그래밍을 하여야 합니다. 그렇기에 이론 정리 노트를 통해 개념과 이론을 학습하였기에 이와 연관 지어 프로그래밍을 시작하기 쉬워 좋았습니다. 그리고 실습 수업에 대한 수업 노트는 밑에서 보시는 바와 같이 저는 수업시간에 했던 사소한 코딩까지 정리하여 저만의 것으로 만들었습니다. 1주차부터 수업이 끝나는 16주차까지 정리했으며, 3개의 과제와 중간, 기말 모두 코드를 직접 짜서 저만의 노트로 정리했습니다. 이 실습 정리 노트는 저 스스로 그때의 기억을 떠올리는 매개체로 사용했을 뿐만 아니라 개요를 보고 제가 스스로 다시 프로그램을 짜서 과거의 프로그램과 비교하였습니다. 이를 통해 과거와 다른 현재 프로그램의 장단점을 분석하고 어떠한 코드가 더 간결하고 전달력 있고 효율적인지 판단해보았습니다. 그리고 저는 프로그래밍 또한 기출이 중요하다고 생각합니다. 시험문제는 매년 바뀌겠지만 내용이 바뀔 뿐이지 저희가 수업시간에 배웠던 다형성, 상속 등과 같은 개념은 계속 사용되기에 수업시간에 학습하거나 과제로 받은 프로그램들을 모아 암기하고 이를 바탕으로 시험문제를 예상해보는 과정을 통해 시험을 준비하였습니다. 이처럼 이론 정리 노트와 실습 정리 노트는 예습 및 복습부터 시험 준비까지 다양하게 사용되기에 객체지향프로그래밍 및 실습을 학습할 때 꼭 필요한 준비물이라 생각합니다.

그리고 객체지향프로그래밍 및 실습 시험 준비를 할 때는 반복이라는 방법을 사용하였습니다. 저는 이 방법을 특이하게 사용하였는데 객체지향프로그래밍 및 실습 과목은 JAVA, 즉 프로그래밍 언어에 대한 이해가 되어야 이론과 실습 문제를 모두 풀 수 있습니다. 그렇기에 코드를 보고 이해를 할 수 있어야 하는데 이는 여러 책과 수업시간에 배운 내용을 반복적으로 학습하는 것을 통해 연습했습니다. 먼저 앞에서 소개한 바와 같이 강의노트부터 수업시간에 배운 내용들과 제가 스스로 조사한 내용들을 집약적으로 요약한 정리 노트에 있는 코드들을 주마다 반복하여 학습하였습니다. 기존의 코드를 스스로 읽어보면서 각각의 코드의 기능을 요약했고, 같은 프로그램을 다른 방법을 통해 만들 수 있는지도 생각하면서 코딩하였습니다. 예를 들어 if문으로 제작한 프로그램을 switch case문을 통해 이를 변형시키거나, enhanced for문을 사용하는 등의 노력부터 시작하여 완전히 다른 기능을 통해 프로그래밍하는 노력을 해보면서 코드를 완벽히 이해할 수 있도록 노력했습니다. 그리고 밑에 있는 사진처럼 저는 여러 책을 참고하면서 공부를 하였습니다. 여러 책을 통하여 공부하였는데 Do it 자바 프로그래밍, Java How to Program, 난 정말 JAVA

를 공부한 적이 없다구요, 명품 자바프로그래밍 등의 책을 통하여 공부하였습니다. 이 많은 책들 중에서 저는 Do it 자바 프로그래밍이 가장 설명이 잘 되어 있다고 생각하고 저는 이 책을 기본으로 생각하고 공부를 진행했습니다. 책 중간중간에 개념을 공부하고 확인할 수 있게 빈칸 넣기 문제들이 많아 공부할 때 유익했고 GUI와 같은 경우에는 아주대학교 도서관에 직접 가서 4층에서 책을 찾아보았는데 많은 책들 중에서 어서와 Java는 처음이지! : JDK8로 배우는 자바 프로그래밍이라는 책이 가장 설명이 잘 되어 있었습니다. 이처럼 다양한 책들을 바탕으로 공부하였기에 수많은 예제들을 접해 시험을 볼 때도 비슷한 유형의 문제들이어서 당황하지 않고 풀 수 있었으며 거의 하루에 2시간 이상씩 매일매일 공부를 지속하였기에 Java 프로그래밍 언어 자체에 대한 이해도 높아졌습니다.

저는 한 학기 동안 객체지향프로그래밍 및 실습 과목 수업에 최선을 다했으며, 연습 및 복습, 이론과 실습 정리 노트, 다양한 책과 자료들을 바탕으로 공부를 하였습니다. 이 밑에 있는 사진들은 제가 공부한 내용들을 정리해놓은 자료들이며, 이러한 수많은 노력을 하고 저만의 학습을 해오면서 큰 보람을 느꼈습니다. 최선을 다한 만큼 후회가 없었고, 이번 기회를 바탕으로 앞으로 공부할 때 이 경험을 기억하며, 최선을 다해 공부하겠습니다. 감사합니다.

★수업시간에 배운 내용들을 요약 및 정리

- "바퀴를 재발명"하는 대신 재사용할 수 있는 다양한 사전 정의된 클래스 세트

- 클래스를 패키지(관련 클래스 그룹)로 분류하고 Java 클래스 라이브러리 또는 Java Application Programming Interface(JAVA API)로 통칭합니다.

- 가져오기 선언을 사용하여 Java 프로그램에 사용되는 미리 정의된 클래스를 식별합니다.

논리 타입 boolean (1비트, true 또는 false) 자바 가장 기본적인 처리되는 boolean의 실제 크기는 1비트가 아닐 수 있다.

문자 타입 char (2바이트, Unicode)

byte (1바이트, -128~127)

정수 타입 short (2바이트, -32768~32767) JDK8부터 양수(0~2¹⁵-1)로도 사용 가능

int (4바이트, -2³¹~2³¹-1) JDK8부터 양수(0~2³¹-1)로도 사용 가능

long (8바이트, -2⁶³~2⁶³-1)

실수 타입 float (4바이트, -3.4E38~3.4E38)

double (8바이트, -1.7E308~1.7E308)

Object Oriented Paradigm

- 실제 문제를 협업 개체로 보는 동시에 절차 지향은 일련의 절차로 봅니다.
- 사용자 관점에 최대한 근접한 현실 모델링
- 익숙한 논리를 사용하여 계산 환경과 쉽게 상호 작용
- 처음부터 모든 것을 레코딩하지 않고도 구성 요소의 구현을 쉽게 수정 및 확장할 수 있습니다.
- ◆ 절차 지향 프로그래밍
- ◆ 작업 순서 표현
- ◆ 작업을 함수로 작성한, 함수들의 집합
- ◆ 객체 지향 프로그래밍
- ◆ 객체들간의 상호 작용으로 표현
- ◆ 클래스 혹은 객체들의 집합으로 프로그램 작성
- ◆ 클래스(class): 객체 모양을 선언한 틀(템플릿)
- ◆ 메소드(멤버 함수)와 필드(멤버 변수)는 모두 클래스 내에 구현
- ◆ 객체
- ◆ 클래스의 모양대로 생성된 실제(instance)
- ◆ 클래스 선언. class Circle
- ◆ class 키워드로 선언
- ◆ 클래스는 [로 시작하여]로 닫으며 이곳에 모든 필드와 메소드 구현
- ◆ class Circle은 Circle 이름의 클래스 선언
- ◆ 클래스 접근 권한. public

부도 클래스 : 슈퍼 클래스 (상위클래스)

자식 클래스 : 서브 클래스 (하위클래스). 슈퍼 클래스를 재사용하고 새로운 특성 추가

class Animal / class Human extend Animal

Polymorphism

- ◆ 다형성
- ◆ 같은 이름의 메소드가 클래스나 객체에 따라 다르게 동작하도록 구현
- ◆ 다형성 사례
- ◆ 메소드 오버로딩 : 같은 이름이지만 다르게 작동하는 여러 메소드
- ◆ 메소드 오버라이딩 : 슈퍼클래스의 메소드를 서브 클래스마다 다르게 구현

```

1 // Fig. 3.1: Account.java
2 // Account class that contains a name instance variable
3 // and methods to set and get its value.
4
5 public class Account {
6     private String name; // instance variable
7
8     // method to set the name in the object
9     public void setName(String name) {
10         this.name = name; // store the name
11     }
12
13     // method to retrieve the name from the object
14     public String getName() {
15         return name; // return value of name to caller
16     }
17 }
                    
```

생성자(Constructor)

- ◆ 객체가 생성될 때 초기화를 위해 실행되는 메소드
- ◆ 생성자의 특징
- ◆ 생성자는 메소드
- ◆ 생성자 이름은 클래스 이름과 반드시 동일
- ◆ 생성자 여러 개 작성 가능 (오버로딩)
- ◆ 생성자는 new를 통해 객체를 생성할 때, 객체당 한 번 호출
- ◆ 생성자는 리턴 타입을 지정할 수 없음
- ◆ 생성자의 목적은 객체 초기화
- ◆ 생성자는 객체가 생성될 때 반드시 호출됨
- ◆ 그러므로 하나 이상 선언되어야 함
- ◆ 개발자가 생성자를 작성하지 않았으면 컴파일러가 자동으로 기본 생성자 삽입

컴파일러
 -소스 코드의 내용은 몇몇 기계 명령에 의해 번역될 수 있다.
 -컴파일러는 모든 소스 코드를 번역해서 기계어 파일을 만든다. 그리고 기계어 파일이 실행되는 것이다.

왜 자바를 이용하나?(Why Java)
 -1995년에 썬 마이크로시스템즈의 제임스 고슬링에 의해 발명되었다.
 -자바의 원칙은 한번 코드가 작성되면 어디서나 실행 가능하다는 것이다. (Write Once Run Anywhere(WORA))

Java Platform
 -자바 플랫폼은 자바 가상 머신(JVM)과 Java API 두가지 요소로 구성되어 있다.
 -자바 가상 머신
 자바 플랫폼의 기초가 되는 요소
 다양한 하드웨어 플랫폼에 사용될 수 있다.
 - Java API
 여러 가지 유용한 기능을 제공하는 기성 소프트웨어 구성 요소의 대규모 컬렉션입니다.
 관련 클래스 및 인터페이스의 라이브러리로 그룹화되며, 이러한 라이브러리를 패키지라고 합니다.

Java의 특징
 ✦ Java Is Simple
 ✦ Java Is Object-Oriented
 ✦ Java Is Distributed
 ✦ Java Is Interpreted
 ✦ Java Is Robust
 ✦ Java Is Secure
 ✦ Java Is Architecture-Neutral
 ✦ Java Is Portable
 ✦ Java's Performance
 ✦ Java Is Multithreaded
 ✦ Java Is Dynamic

Java 파일의 실행 과정 5단계
 1. edit : 소스 프로그램을 Java로 작성
 2. compile : class 확장자를 가진 바이트코드 프로그램 생성
 RUN(load, verify, execute)
 3. load : class loader가 .class 파일들을 primary memory로 load하고, 실행할 프로그램에 필요한 다른 .class 파일도 load한다.
 4. verify : bytecode verifier가 load된 class파일들의 바이트코드들을 보안을 위해 검사한다.

5. execute : JVM이 바이트코드를 읽어내 실행한다. 성능향상을 위해 JIT 컴파일러가 존재.

Bytecodes Disassemble
 ✦ 클래스 파일에 들어 있는 바이트 코드를 텍스트로 볼 수 있게 변환하는 작업
 ✦ JDK의 javap.exe 이용

Java Development and Execution Environment
 ✦ 자바 프로그램의 개발
 • 여러 소스(.java)로 나누어 개발
 • 바이트 코드(.class)를 하나의 실행 파일(exe)로 만드는 링크 과정 없음
 ✦ 실행
 • main() 메소드를 가진 클래스에서 부터 실행 시작
 • 자바 가상 기계는 필요할 때, 클래스 파일 로딩, 적은 메모리로 실행 가능

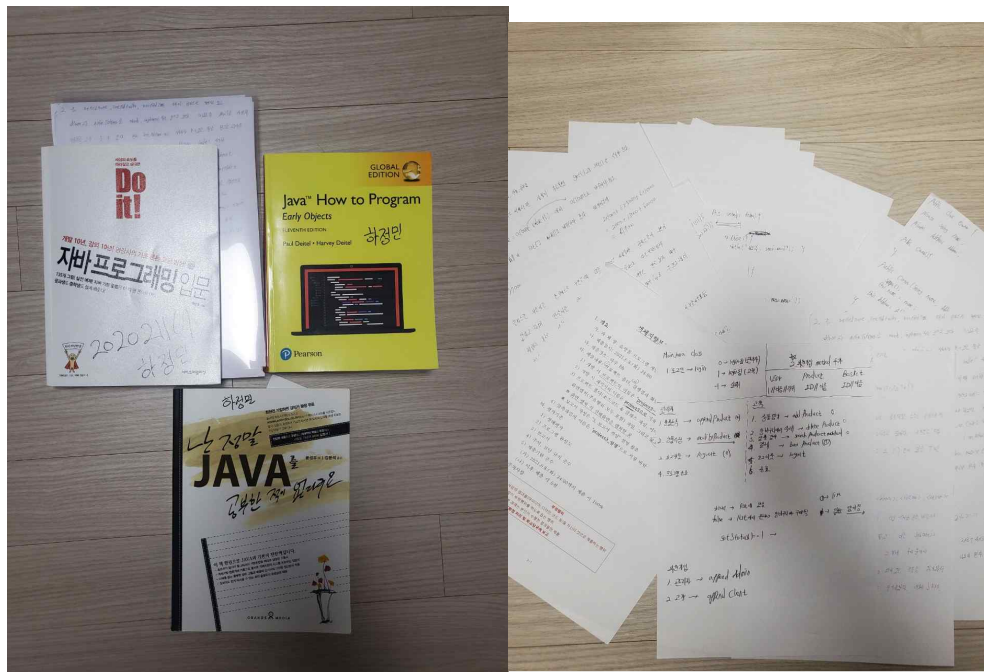
Class
 -모든 자바 프로그램은 적어도 하나의 클래스가 존재한다.
 -class 키워드는 class 선언을 뜻하며, class 뒤에 붙는 예가 클래스의 이름이다.
 -키워드는 모두 소문자로 이루어져 있으며, 프로그래밍을 하기 위한 예약어이다.
 -public class의 이름은 파일이름과 같아야 한다. 그래서 test.java 파일명이어야 한다.
 -전통적으로 클래스 이름은 첫 문자는 대문자로 시작하고 각각의 단어가 새로 시작될 때 대문자를 사용한다.
 -클래스 이름은 식별자이다. 숫자로 시작하면 안되며, 공백이 포함되면 안된다.
 -자바는 대소문자 구분한다.

Method
 - 모든 Java 애플리케이션의 시작점.
 - 식별자 후 뒤에 괄호 안에 method라는 프로그래밍 빌드 블록이 있습니다.
 -Java 클래스 선언에는 일반적으로 하나 이상의 메서드가 포함됩니다.
 - main은 그림과 같이 정의되어야 합니다. 그렇지 않으면 JVM이 애플리케이션을 실행하지 않습니다.
 - 작업 수행 방법 및 작업 완료 시 정보를 반환할 수 있습니다.
 - 키워드 void는 이 방법이 정보를 반환하지 않음을 나타냅니다.
 - public static void main = static public 가능

✦ System.out.printf method
 • f means "formatted"
 • displays formatted data
 • system.out.printf("%s", "Welcome");

Import
 - Import는 컴파일러가 이 프로그램에 사용되는 클래스를 찾는 데 도움이 됩니다.

★수업시간 이후 예습 복습을 위한 학습 자료들



★주차별로 정리하여 학습한 코드들

1주차

```
package week01;
```

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("학번_이름");
    }
}
```

2주차-1

```
package week02;
```

```
public class EnhancedForExample {
    public static void main(String[] args) {
        int scores[] = {95, 71, 84, 93, 87};
        int sum = 0;

        for(int i = 0 ; i < scores.length ; i++ ) {
            sum += scores[i];
        }

        System.out.print("합:" + sum + " 평균:" + sum/(scores.length));
    }
}
//for(int tmp : scores) {
// sum += tmp;
package week02;
```

```
public class SwitchStringEx {
    public static void main(String[] args) {
        String position = "과장";

        switch(position) {
            case "부장":
```

```

        System.out.println("700만원");
        break;
    case "과장":
        System.out.println("500만원");
        break;
    default:
        System.out.println("300만원");
        break;
    }
}
}
}

```

2주차-2

```

package hw02;

import java.util.Scanner;

public class FindSubject {
    public static void main(String[] args) {
        String[] subject = new String[] {"수학1",
        "객체지향프로그래밍및실습", "항공력의역할", "군사탐방"};
        String input;
        Scanner sc = new Scanner(System.in);
        System.out.print("수강 신청할 과목명을 입력하세요: ");
        input = sc.next();

        // 코드 작성 부분

        int subject1[] = new int [8];

        int sum, i=0;

        for(String arr : subject)

```

```
    {  
        if(arr.equals(input) == true)  
        {  
            subject1[i] = 1;  
            i++;  
        }  
        else  
        {  
            subject1[i] = 0;  
            i++;  
        }  
    }  
  
    sum = subject1[0] + subject1[1] + subject1[2] + subject1[3];  
  
    if(sum>0)  
    {  
        System.out.println("해당 과목이 존재합니다.");  
    }  
    else  
        System.out.println("해당 과목이 존재하지 않습니다.");  
    }  
}
```

3주차-1

```
package week03;
```

```
public class Polygon {  
    int point;  
    double angle;  
  
    public Polygon(int point) {  
        this.point = point;  
    }  
}
```

```
public void printAngle() {
    angle = 180.0*(point-2)/point;
    System.out.printf("정%d각형의 한 내각의 크기는 %f입니다.\n", point, angle);
    return;
}
}
```

```
package week03;
```

```
import java.util.Scanner;
```

```
public class PolygonTest {
    public static void main(String[] args) {
        int point;
        Scanner input = new Scanner(System.in);

        System.out.println("꼭짓점 개수를 입력해 주세요.: ");
        point = input.nextInt();

        Polygon poly = new Polygon(point);

        poly.printAngle();
    }
}
```

```
package week03;
```

```
public class Student {
    String name;

    public Student(String name) {
        this.name = name;
    }
}
```

```
    public void raiseHand() {  
        System.out.println(name + “이(가) 손을 듭니다.”);  
    }  
}
```

```
package week03;
```

```
public class StudentExample {  
    public static void main(String[] args) {  
        Student s1 = new Student(“김국디”);  
        Student s2 = new Student(“이아주”);  
  
        s1.raiseHand();  
        s2.raiseHand();  
    }  
}
```

3주차-2

```
package hw03;
```

```
public class Triangle {  
    double area, area1, area2;  
    int height, width;  
  
    public Triangle()  
    {  
        this.height = height;  
        this.width = width;  
    }  
  
    public double calculateArea(int height, int width)  
    {  
        area = height*width / 2.0;  
        return area;  
    }  
}
```

```
}

public static void compareArea(double area1, double area2)
{
    System.out.printf("첫번째 삼각형의 넓이는 %f입니다.\n", area1);
    System.out.printf("두번째 삼각형의 넓이는 %f입니다.\n", area2);
    System.out.printf("가장 큰 넓이는 %f입니다.\n", area1 > area2 ? area1 :
area2);
}
}

package hw03;

import java.util.Scanner;

public class TriangleTest {
    public static void main(String[] args) {
        int height, width;
        double area, area1, area2;

        Scanner input = new Scanner(System.in);
        Triangle tri = new Triangle();

        System.out.println("첫번째 삼각형의 높이와 너비를 입력하세요.");
        height = input.nextInt();
        width = input.nextInt();

        while(true)
        {
            if((height<=0)||width<=0)
            {
                System.out.println("입력값이 음수가 될 수 없습니다.");

                System.out.println("첫번째 삼각형의 높이와 너비를 입력하세요.");
                height = input.nextInt();
            }
        }
    }
}
```

```

        width = input.nextInt();
    }
    else
        break;
}

area1 = tri.calculateArea(height, width);

System.out.println("두번째 삼각형의 높이와 너비를 입력하세요.");
height = input.nextInt();
width = input.nextInt();

while(true)
{
    if((height<=0)||(width<=0))
    {
        System.out.println("입력값이 음수가 될 수 없습니다.");

        System.out.println("두번째 삼각형의 높이와 너비를 입력하세요.");
        height = input.nextInt();
        width = input.nextInt();
    }
    else
        break;
}
area2 = tri.calculateArea(height, width);

Triangle.compareArea(area1, area2);
}
}

```

4주차-1

```
package week04;
```

```
public class BitCoin {
```

```
private String name;
private double price;
private int number;

public BitCoin(String name, double price, int number)
{
    setName(name);
    setPrice(price);
    setNumber(number);
}

public double calcPrice() {
    return price*number;
}

public String toString() {
    return "Coin 이름: " + getName() + ", 가격: " + getPrice() + ",
개수: " + getNumber() + ", 총 가격: " + calcPrice();
}

public String getName() {
    return name;
}

public void setName(String name)
{
    this.name = name;
}

public double getPrice() {
    return price;
}

public void setPrice(double price)
{
```

```
        this.price = price;
    }

    public int getNumber()
    {
        return number;
    }

    public void setNumber(int number)
    {
        this.number = number;
    }
}

package week04;

import java.util.ArrayList;

public class BitCoinTest {
    public static void main(String[] args) {
        ArrayList<BitCoin> myWallet = new ArrayList<>();
        myWallet.add(new BitCoin("BTC", 9543.66, 10));
        myWallet.add(new BitCoin("ETH", 3240.17, 20));
        myWallet.add(new BitCoin("XRP", 910.95, 10));
        myWallet.add(new BitCoin("LTC", 1184.12, 30));

        for(BitCoin b : myWallet)
        {
            System.out.println(b.toString());
        }
    }
}
```

```
package week04;

public class Car {
    private int speed;
    private boolean stop;

    public int getSpeed() {
        return speed;
    }

    public void setSpeed(int speed)
    {
        if(speed < 0) {
            this.speed = 0;
            return;
        }
        else {
            this.speed = speed;
            return;
        }
    }

    public boolean isStop() {
        return stop;
    }

    public void setStop(boolean stop) {
        this.stop = stop;
    }
}
```

```
package week04;

public class CarTest {
    public static void main(String[] args) {
```

```

Car myCar = new Car();

myCar.setSpeed(-50);
System.out.println("현재 속도: " + myCar.getSpeed());

myCar.setSpeed(60);
System.out.println("현재 속도: " + myCar.getSpeed());

if(!myCar.isStop()) {
    myCar.setStop(true);
}
System.out.println("현재 속도: " + myCar.getSpeed());
}
}

```

4주차-2

```
package hw04;
```

```
public class Account
{
```

```
    private String accountNumber; //계좌번호
```

```
    private String accountowner; //계좌주
```

```
    private int balance; //잔액
```

```
// private으로 설정
```

```
    public Account(String accountNumber, String accountowner, int balance)
```

```
{
```

```
        setAccountNumber(accountNumber);
```

```
        setAccountowner(accountowner);
```

```
        setBalance(balance);
```

```
}
```

```
// toString() method 사용
```

```
    public String toString()
```

```
{
```

```
        return getAccountNumber() + " " + getAccountowner() + " " +
getBalance();
    }

    // Getter, Setter 사용
    public String getAccountNumber()
    {
        return accountNumber;
    }

    public void setAccountNumber(String accountNumber)
    {
        this.accountNumber = accountNumber;
    }

    public String getAccountowner()
    {
        return accountowner;
    }

    public void setAccountowner(String accountowner)
    {
        this.accountowner = accountowner;
    }

    public int getBalance()
    {
        return balance;
    }

    public void setBalance(int balance)
    {
        this.balance = balance;
    }
}
```

```
package hw04;

import java.util.ArrayList;
import java.util.Scanner; // 입력

public class Bank
{
    private ArrayList<Account> ab = new ArrayList<>(); // 생성

    Scanner a = new Scanner(System.in);

    public void makeAccount() //계좌 생성
    {
        System.out.print("계좌번호: ");
        String accountNumber = a.next();

        System.out.print("계좌주: ");
        String accountowner = a.next();

        System.out.print("초기입금액: ");
        int balance = a.nextInt();

        for (Account data : ab)
        {
            if (data.getAccountNumber().equals(accountNumber))
            {
                System.out.println("이미 존재하는 계좌입니다. 계좌 생성 오류");
                return;
            }
        }

        if (balance <= 0)
        {
            System.out.println("초기 입금액은 0원 이상이어야 합니다.");
        }
    }
}
```

```
        return;
    }

    Account account = new Account(accountNumber, accountowner, balance);
    ab.add(account);

    System.out.println("결과: 계좌가 생성되었습니다.");
}

public void seeAccount() //계좌조회
{
    if (ab.size() == 0)
    {
        System.out.println("개설 계좌가 존재하지 않습니다.");
    }
    else
    {
        int totalBalance = 0;

        for (Account account : ab)
        {
            totalBalance += account.getBalance();
            System.out.println(account.toString());
        }
        System.out.println("총 잔고: " + totalBalance);
    }
}

public void deposit() // 예금
{
    System.out.print("계좌번호: ");
    String accountNumber = a.next();

    System.out.print("예금액: ");
    int balance = a.nextInt();
}
```

```
if (balance <= 0)
{
    System.out.println("예금액이 0원 이상이어야 합니다. 예금 실패.");
    return;
}

for (Account account : ab)
{
    if (account.getAccountNumber().equals(accountNumber))
    {
        account.setBalance(account.getBalance() + balance);
        System.out.println("예금이 성공되었습니다.");
        return;
    }
}
System.out.println("해당 계좌가 존재하지 않습니다. 예금 실패.");
}

public void withdraw() // 출금
{
    System.out.print("계좌번호: ");
    String accountNumber = a.next();

    System.out.print("출금액: ");
    int balance = a.nextInt();

    if (balance <= 0)
    {
        System.out.println("출금액이 0원 이상이어야 합니다. 출금 실패.");
        return;
    }

    for (Account account : ab)
    {
```

```

        if (account.getAccountNumber().equals(accountNumber))
        {
            account.setBalance(account.getBalance() - balance);
            System.out.println("출금이 성공되었습니다.");
            return;
        }
    }
    System.out.println("해당 계좌가 존재하지 않습니다. 출금 실패.");
}
}

```

```
package hw04;
```

```
import java.util.Scanner;
```

```
public class BankTest
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        Scanner a = new Scanner(System.in);
```

```
        Bank b = new Bank();
```

```
        int answer;
```

```
        do
```

```
        {
```

```
            System.out.println("-----");
```

```
            System.out.println("1.계좌생성 | 2.계좌조회 | 3.예금 | 4.출금 | 5.종료");
```

```
            System.out.println("-----");
```

```
            System.out.print("선택: ");
```

```
            answer = a.nextInt();
```

```
            switch (answer)
```

```
            {
```

case 1:

```
System.out.println("-----");
System.out.println("계좌생성");
System.out.println("-----");
b.makeAccount();
break;
```

case 2:

```
System.out.println("-----");
System.out.println("계좌조회");
System.out.println("-----");
b.seeAccount();
break;
```

case 3:

```
System.out.println("-----");
System.out.println("  예금  ");
System.out.println("-----");
b.deposit();
break;
```

case 4:

```
System.out.println("-----");
System.out.println("  출금  ");
System.out.println("-----");
b.withdraw();
break;
```

case 5:

```
System.out.println("프로그램 종료");
break;
```

default:

```
System.out.println("올바른 입력 값을 넣어주세요.");
}
```

```
    } while (answer != 5);  
  
    }  
  
}
```

5주차-1

```
package week05;
```

```
public class Box {  
    private int width;  
    private int length;  
    private int height;  
  
    public Box(int width, int length, int height) {  
        this.width = width;  
        this.length = length;  
        this.height = height;  
    }  
  
    public String toString() {  
        return "폭: " + width + ", 너비: " + length + ", 높이: " + height;  
    }  
  
    public boolean isSameBox(Box b) {  
        if(b.width == this.width && b.length == this.length && b.height ==  
this.height) {  
            return true;  
        }  
        else return false;  
    }  
  
}
```

```
package week05;
```

```
public class BoxTest {  
  
    public static void main(String[] args) {  
        Box box1 = new Box(10, 20, 30);  
        Box box2 = new Box(10, 20, 30);  
  
        System.out.println(box1);  
        System.out.println(box2);  
  
        if(box1.isSameBox(box2)) System.out.println("두 박스는 동일한  
치수입니다.");  
    }  
}
```

```
package week05;
```

```
public class Date {  
    private int year;  
    private String month;  
    private int day;  
  
    public Date() {  
        this(1900, "1월", 1);  
    }  
  
    public Date(int year) {  
        this(year, "1월", 1);  
    }  
  
    public Date(int year, String month, int day) {  
        this.year = year;  
        this.month = month;  
        this.day = day;  
    }  
}
```

```

    public String toString() {
        return "Date [year: " + year + ", month: " + month + ", day: " +
day + "];
    }
}

```

```
package week05;
```

```

public class DataTest {
    public static void main(String[] args) {
        Date date1 = new Date(2015, "8월", 10);
        Date date2 = new Date(2021);
        Date date3 = new Date();

        System.out.println(date1);
        System.out.println(date2); // toString 연결된 method 호출
        System.out.println(date3);
    }
}

```

5주차-2

```
package hw05;
```

```

public class Admin {
    private String id;
    private String pw;
    public Admin(String id, String pw)
    {
        this.setId(id);
        this.setPw(pw);
    }
    public String getId() {return id;}
    public void setId(String id) {this.id = id;}
}

```

```

    public String getPw() {return pw;}
    public void setPw(String pw) {this.pw = pw;}
}

package hw05;

public class Book {

    private int id;//도서의 등록번호
    private boolean status = true;// 도서 상태
    private String title; // 도서명
    public Book(int id, boolean status, String title)
    {
        this.setId(id);
        this.setStatus(status);
        this.setTitle(title);
    }

    public int getId() {return id;}
    public void setId(int id) {this.id = id;}
    public boolean isStatus() {return status;}
    public void setStatus(boolean status) {this.status = status;}
    public String getTitle() {return title;}
    public void setTitle(String title) {this.title = title;}
}

package hw05;
import java.util.ArrayList;

public class Library {
    ArrayList<Book> bookList = new ArrayList<Book>(); //도서관에 있는 도서들
    ArrayList<Admin> adminList = new ArrayList<Admin>(); //관리자 목록
    ArrayList<Student> studentList = new ArrayList<Student>();//등록된 학생목록
    public boolean insertStudent(String id, String pw) //학생 등록
    {
        for (Student student : studentList)

```

```
{
    if (student.getId().equals(id))
    {
        return false;
    }
}
studentList.add(new Student(id, pw));
return true;
}
public boolean deleteStudent(String id) //학생 삭제
{
    for (Student student : studentList)
    {
        if (student.getId().equals(id))
        {
            if (student.getBorrowList().isEmpty())
            {
                studentList.remove(student);
                return true;
            }
        }
    }
    return false;
}
public Book searchBook(int id, String title) //책 검색
{
    for (Book book : bookList)
    {
        if (book.getId() == id && book.getTitle().equals(title))
        {
            return book;
        }
    }
    return null;
}
```

```

    public boolean borrowBook(Student student, int bookId, String
bookTitle)//도서대출
    {
        Book book = searchBook(bookId, bookTitle);
        if (book != null)
        {
            return student.borrowBook(book);
        }
        return false;
    }

    public boolean returnBook(Student student, int bookId, String bookTitle) //도서
반납
    {
        Book book = searchBook(bookId, bookTitle);
        if (book != null) {
            return student.returnBook(book);
        }
        return false;
    }

    public Admin adminLogin(String id, String pw) //로그인 관리자
    {
        for (Admin admin : adminList)
        {
            if (admin.getId().equals(id) && admin.getPw().equals(pw))
            {
                return admin;
            }
        }
        return null;
    }

    public Student studentLogin(String id, String pw) // 학생 로그인
    {
        for (Student student : studentList)
        {
            if (student.getId().equals(id) && student.getPw().equals(pw))

```

```
        {  
            return student;  
        }  
    }  
    return null;  
}  
}
```

```
package hw05;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class Student {  
    private String id;//학생의 학번  
    private String pw;//학생의 비밀번호  
    private List<Book> borrowList;//학생이 빌린 도서  
    public Student(String id, String pw)  
    {  
        this.id = id;  
        this.pw = pw;  
        this.borrowList = new ArrayList<>();  
    }  
    public boolean borrowBook(Book newBook) //도서 대출 method  
    {  
        for (Book book : borrowList)  
        {  
            if (book.getId() == newBook.getId())  
            {  
                return false;  
            }  
        }  
        borrowList.add(newBook);  
        return true;  
    }  
}
```

```

public boolean returnBook(Book bookToReturn) //도서반납 method
{
    for (Book book : borrowList)
    {
        if (book.getId() == bookToReturn.getId())
        {
            borrowList.remove(book);
            return true;
        }
    }
    return false;
}

public String getId() {return id;}
public void setId(String id) {this.id = id;}
public String getPw() {return pw;}
public void setPw(String pw) {this.pw = pw;}
public List<Book> getBorrowList() {return borrowList;}
}

```

```
package hw05;
```

```
import java.util.Scanner;
```

```

public class LibraryTest {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Library library = new Library();
        library.bookList.add(new Book(111, true, "book1"));
        library.bookList.add(new Book(222, true, "book2"));
        library.bookList.add(new Book(333, true, "book3"));
        library.bookList.add(new Book(444, true, "book4"));
        library.bookList.add(new Book(555, true, "book5"));
        library.adminList.add(new Admin("12345", "mdc2021")); //포 활용
        Admin admin = null;
        Student student = null;
    }
}

```

```
int lm = 0;
int sm;
int am;
while (lm != 3) //메뉴 화면
{
    if (admin == null && student == null)
    {
        System.out.println("1. 학생 로그인");
        System.out.println("2. 관리자 로그인");
        System.out.println("3. 프로그램 종료");
        lm = s.nextInt();
        String id, pw;
        switch (lm) //switch case문으로 각 항목에 해당하는 경우 출력
        {
            case 1://로그인
                System.out.print("ID 입력\n");
                id = s.next();
                System.out.print("PW 입력\n");
                pw = s.next();
                student = library.studentLogin(id, pw);
                if (student == null)
                {
                    System.out.println("로그인 실패");
                }
                break;
            case 2://로그인
                System.out.print("ID 입력\n");
                id = s.next();
                System.out.print("PW입력\n");
                pw = s.next();
                admin = library.adminLogin(id, pw);
                if (admin == null)
                {
                    System.out.println("로그인 실패");
                }
        }
    }
}
```

```

        break;
    }
}
else if (student != null) //학생
{
    System.out.println("1. 도서 대출");
    System.out.println("2. 도서 반납");
    System.out.println("3. 로그 아웃");
    sm = s.nextInt();
    int bookId;
    String bookTitle;
    switch (sm)
    {
        case 1://대출
            System.out.print("대출을 원하시는 도서의 등록번호
입력\n");

            bookId = s.nextInt();
            System.out.print("대출을 원하시는 도서명 입력\n");
            bookTitle = s.next();
            if (library.borrowBook(student, bookId, bookTitle))
            {
                System.out.println("대출 성공");
            }
            else
            {
                System.out.println("대출 실패");
            }
            break;
        case 2://반납
            System.out.print("반납을 원하시는 도서의 등록번호
입력\n");

            bookId = s.nextInt();
            System.out.print("반납을 원하시는 도서명 입력\n");
            bookTitle = s.next();
            if (library.returnBook(student, bookId, bookTitle))

```

```
        {
            System.out.println("반납 성공");
        }
        else
        {
            System.out.println("반납 실패");
        }
        break;
    case 3://로그아웃
        student = null;
        System.out.println("로그아웃");
        break;
    }
}
else //관리자
{
    System.out.println("1. 학생 등록");
    System.out.println("2. 학생 삭제");
    System.out.println("3. 로그 아웃");
    am = s.nextInt();
    String id, pw;
    switch (am)
    {
        case 1://등록
            System.out.print("학생 학번 입력\n");
            id = s.next();
            System.out.print("학생 비번 입력\n");
            pw = s.next();
            if (library.insertStudent(id, pw))
            {
                System.out.println("등록 성공");
            }
            else
            {
                System.out.println("등록 실패");
            }
        }
    }
}
```

```

    }
    break;
case 2://삭제
    System.out.print("학생 학번 입력\n");
    id = s.next();
    if (library.deleteStudent(id))
    {
        System.out.println("삭제 성공");
    }
    else
    {
        System.out.println("삭제 실패");
    }
    break;
case 3://로그아웃
    admin = null;
    System.out.println("로그아웃");
    break;
    }
    }
    }
}

```

6주차-1

```
package week06;
```

```
public class Child extends Parent {
```

```
}
```

```
package week06;
```

```
public class Parent {
```

```
}
```

```
package week06;
```

```
public class InstanceofExample {
```

```
    public static void method1(Parent p) {  
        if(p instanceof Child) {  
            Child c = (Child) p;  
            System.out.println("method1 - child로 변환 성공");  
        }  
        else {  
            System.out.println("method1 - child로 변환 실패");  
        }  
    }  
}
```

```
    public static void method2(Parent p) {  
        Child c = (Child) p;  
        System.out.println("method2 - child로 변환 성공");  
    }  
}
```

```
    public static void main(String[] args) {  
        Parent parentA = new Child();  
  
        method1(parentA);  
        method2(parentA);  
  
        Parent parentB = new Parent();  
  
        method1(parentB);  
        method2(parentB);  
    }  
}
```

```
    }  
}  
  
package week06;  
  
public class Circle extends Shape{  
    private int radius;  
  
    public void draw() {  
        System.out.println("Circle draw");  
    }  
}
```

```
package week06;  
  
public class Triangle extends Shape{  
    private int base, height;  
  
    public void draw() {  
        System.out.println("Triangle draw");  
    }  
}
```

```
package week06;  
  
public class Rectangle extends Shape{  
    private int width, height;  
  
    public void draw() {  
        System.out.println("Rectangle draw");  
    }  
}
```

```
package week06;
```

```
public class Shape {  
    protected int x, y;  
  
    public void draw() {  
        System.out.println("Shape draw");  
    }  
}
```

```
package week06;
```

```
import java.util.ArrayList;
```

```
public class ShapeTest {  
  
    public static void main(String[] args) {  
        ArrayList<Shape> figure = new ArrayList<>();  
  
        Shape s1 = new Rectangle();  
        Shape s2 = new Triangle();  
        Shape s3 = new Circle();  
  
        figure.add(s1);  
        figure.add(s2);  
        figure.add(s3);  
  
        for (Shape s : figure) s.draw();  
    }  
}
```

6주차-2

```
package hw06;

public class Member {

    protected String clientname;//고객이름
    protected String contactinfo;//연락처
    protected int buycount;//구매개수
    String clientgrade;
    private double so;
    protected int se;
    public Member(String name, String number, int count)//생성자
    {
        this.clientname = name;
        this.contactinfo = number;
        this.buycount = count;
    }
    public Member() {};
    static int p = 1000;
    public double calcSales(int p, double so)//calcSales 메소드
    {
        if (clientgrade == null)
        {
            System.out.println("등급이 정해지지 않았습니다");
        }
        se = (int)(p * so);
        return 0;
    }
    public double getSo() {
        return so;
    }
    public void setSo(double so) {
        this.so = so;
    }
}
```

```
package hw06;
```

```
public class BronzeMember extends Member {
    public BronzeMember(String name, String number, int count) //생성자
    {
        super(name, number, count);
    }
    public BronzeMember() {};
    public double calcSales(int buycount) //override
    {
        se = (int)p;
        return se*buycount;
    }
}
```

```
package hw06;
```

```
public class SilverMember extends Member {
    public SilverMember(String name, String number, int buycount, double so)
    {
        super(name, number, buycount);
    }
    public SilverMember() {};
    private double so = 0.9;//할인율
    public double calcSales(int buycount) //override
    {
        se = (int)(so*p);
        return se *buycount;
    }
}
```

```
package hw06;
```

```
public class GoldMember extends Member {
```

```
public GoldMember(String name, String number, int buycount, double so)
{
    super(name, number, buycount);
}
public GoldMember() {};
private double so = 0.5;
public double calcSales(int buycount) //override
{
    se=(int)(p * so);
    return se *buycount;
}
}
```

```
package hw06;
```

```
import java.util.Scanner;
import java.util.ArrayList;
```

```
public class MemberTest {
    Scanner sc = new Scanner(System.in);
    static double sum;
    public static void createMember(String name, String number, int buycount)
    {
        ArrayList<Member> m = new ArrayList<Member>();//arraylist
        Scanner sc = new Scanner(System.in);
        m.add(new Member(name, number, buycount));
    }
    public static String calcSum() //총 매출
    {
        String Stringssum;
        Stringssum = Double.toString(sum);
        return Stringssum;
    }
    public static void addsum(double i)
    {
```

```
        sum += i;
    }
    public static void main(String[] args)
    {
        String name;
        String number;
        int buycount;
        int c = 0;
        while (true) // 반복
        {
            System.out.println("-----");
            System.out.println("1. Bronze 등급 고객 등록 ");
            System.out.println("2. Silver 등급 고객 등록 ");
            System.out.println("3. Gold 등급 고객 등록 ");
            System.out.println("4. 총 매출액 정산 ");
            System.out.println("5. 프로그램 종료 ");
            System.out.println("-----");
            System.out.println("메뉴를 선택하세요: ");
            Scanner sc = new Scanner(System.in);
            c = sc.nextInt();
            if (c == 1)
            {
                System.out.println("고객 이름을 입력하세요: ");
                name = sc.next();
                System.out.println("고객 연락처를 입력하세요: ");
                number = sc.next();
                System.out.println("구매 개수를 입력하세요: ");
                buycount = sc.nextInt();
                BronzeMember bm = new BronzeMember();
                addsum(bm.calcSales(buycount));
                createMember(name, number, buycount);
                System.out.println("정상적으로 등록되었습니다.");
            }
            else if (c == 2)
            {
```

```

System.out.println("고객 이름을 입력하세요: ");
name = sc.next();
System.out.println("고객 연락처를 입력하세요: ");
number = sc.next();
System.out.println("구매 개수를 입력하세요: ");
buycount = sc.nextInt();
SilverMember sm = new SilverMember();
addsum(sm.calcSales(buycount));
createMember(name, number, buycount);
System.out.println("정상적으로 등록되었습니다.");
}
else if (c == 3)
{
System.out.println("고객 이름을 입력하세요: ");
name = sc.next();
System.out.println("고객 연락처를 입력하세요: ");
number = sc.next();
System.out.println("구매 개수를 입력하세요: ");
buycount = sc.nextInt();
GoldMember gm = new GoldMember();
addsum(gm.calcSales(buycount));
createMember(name, number, buycount);
System.out.println("정상적으로 등록되었습니다.");
}
else if (c == 4)
{
System.out.println("총 매출은: " + calcSum() +
"원");
}
else if (c == 5)
{
System.out.println("프로그램을 종료합니다.");
System.exit(0);
}
else

```

```
        {  
            System.out.println("잘못된 입력입니다.");  
        }  
    }  
}
```

7주차-1

```
package week07;
```

```
public class Cat extends Animal{  
    public Cat(String name) {  
        this.name = name;  
    }  
  
    public void sound() {  
        System.out.println("야옹");  
    }  
}
```

```
package week07;
```

```
public class Dog extends Animal{  
    public Dog(String name) {  
        this.name = name;  
    }  
  
    public void sound() {  
        System.out.println("멍멍");  
    }  
}
```

```
package week07;
```

```
public abstract class Animal {  
    public String name;
```

```
        public void breathe() {  
            System.out.println(name + “가 숨을 쉽니다.”);  
        }  
    }  
}
```

```
        public abstract void sound();  
    }  
}
```

```
package week07;
```

```
import java.util.ArrayList;
```

```
public abstract class AnimalTest {
```

```
    public static void main(String[] args) {  
        ArrayList<Animal> animalList = new ArrayList<>();
```

```
        animalList.add(new Cat(“점박이”));  
        animalList.add(new Dog(“왕눈이”));
```

```
        for(Animal a : animalList) a.breathe();
```

```
        for(Animal a : animalList) a.sound();  
    }  
}
```

```
package week07;
```

```
public class Taxi implements Vehicle {  
    public void run() {
```

```
        System.out.println("택시가 달립니다.");
    }
}
```

```
package week07;
```

```
public class Bus implements Vehicle{
    public void run() {
        System.out.println("버스가 달립니다.");
    }
}
```

```
package week07;
```

```
public class Driver {
    public void drive(Vehicle v) {
        v.run();
    }
}
```

```
package week07;
```

```
public interface Vehicle {
    public abstract void run();
}
```

```
package week07;
```

```
public class VehicleTest {

    public static void main(String[] args) {
        Driver driver = new Driver();

        driver.drive(new Bus());
        driver.drive(new Taxi());
    }
}
```

```
    }  
  
}  
  
package week07;  
  
public interface Wearable {  
    public abstract void putOn();  
    public abstract void putOff();  
}  
  
package week07;  
  
public interface Player {  
    void play();  
    void stop();  
}  
  
package week07;  
  
public interface ExPlayer extends Player {  
    void slow();  
}  
  
package week07;  
  
public class VRPlayer implements ExPlayer, Wearable {  
  
    @Override  
    public void play() {  
        // TODO Auto-generated method stub  
        System.out.println("영상을 재생합니다.");  
    }  
}
```

```
@Override
public void stop() {
    // TODO Auto-generated method stub
    System.out.println("영상을 멈춥니다.");
}

@Override
public void putOn() {
    // TODO Auto-generated method stub
    System.out.println("기기를 착용합니다.");
}

@Override
public void putOff() {
    // TODO Auto-generated method stub
    System.out.println("기기를 벗습니다.");
}

@Override
public void slow() {
    // TODO Auto-generated method stub
    System.out.println("영상을 느리게 재생합니다.");
}
}

package week07;

public class VRTester {

    public static void main(String[] args) {
```

```
// TODO Auto-generated method stub
VRPlayer v = new VRPlayer();

v.putOn();
v.play();
v.slow();
v.stop();
v.putOff();

}

}
```

7주차-2

```
package hw07;
```

```
public interface Skinnable
```

```
{
    int black = 0;
    int red = 1;
    int green = 2;
    int blue = 3;
    int yellow = 4;

    void changeSkin(int s);
    void printSkin();
}
```

```
package hw07;
```

```
public class Pet {
    private String name;
    private String masterName;

    public String getName() {
```

```
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getMasterName() {
        return masterName;
    }

    public void setMasterName(String masterName) {
        this.masterName = masterName;
    }

    public Pet(String name, String masterName) {
        this.name = name;
        this.masterName = masterName;
    }

    public void introduce()
    {
        System.out.println("* 내 이름은 " + name + " 입니다.");
        System.out.println("* 주인의 이름은 " + masterName + " 입니다.");
    }
}

package hw07;

public class RobotPet extends Pet{

    public RobotPet(String name, String masterName)
    {
        super(name, masterName);
    }
}
```

```
    }

    public void introduce()
    {
        System.out.println("** 나는 로봇! 내 이름은 " + getName() + "
입니다.");
        System.out.println("** 주인의 이름은 " + getMasterName() + "
입니다.");
    }

    public void work(int select)
    {
        if(select == 1) System.out.println("청소를 합니다.");
        else if(select == 2) System.out.println("세탁을 합니다.");
        else if(select == 3) System.out.println("불을 끕니다.");
    }
}

package hw07;

import java.util.Random;

public class SkinnableRobotPet extends RobotPet implements Skinnable{

    private int skin;

    Random r = new Random();

    public int getSkin() {
        return skin;
    }

    public void setSkin(int skin) {
        this.skin = skin;
    }
}
```

```
public SkinnableRobotPet(String name, String masterName, int skin)
{
    super(name, masterName);
    this.skin = skin;
}

public void changeSkin(int s) {
    skin = r.nextInt(5);
}

public void printSkin()
{
    String c = "초록색";
    if(skin == black) c = "검은색";
    else if(skin == red) c = "빨간색";
    else if(skin == green) c = "초록색";
    else if(skin == blue) c = "파랑색";
    else if(skin == yellow) c = "노랑색";
    System.out.println("Skin은 " + c + " 입니다.");
}
}

package hw07;

import java.util.ArrayList;
import java.util.Random;

public class PetsTest {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int a;
        ArrayList<Pet> p = new ArrayList<>();
```

```

Random r = new Random();

p.add(new Pet("피카츄", "김국디"));
p.add(new RobotPet("꼬북이", "이아주"));
p.add(new SkinnableRobotPet("파이리", "박연암", 3));

for (a = 0; a < 3; a++)
{
    for (Pet pet : p)
    {
        pet.introduce();

        if (pet instanceof RobotPet)
        {
            ((RobotPet) pet).work(1 + r.nextInt(3));
        }

        if (pet instanceof SkinnableRobotPet)
        {
            ((SkinnableRobotPet) pet).changeSkin(r.nextInt(5));
            ((SkinnableRobotPet) pet).printSkin();
        }
    }
    System.out.println();
}
}
}

```

중간과제

```

package project1;

public class Admin {

```

```

private int id; //관리자의 고유 id
private String name; //관리자 이름

public Admin(int id, String name)
{
    this.setId(id);
    this.setName(name);
} // getter와 setter 사용하기

public int getId() {return id;}
public void setId(int id) {this.id = id;}
//id
public String getName() {return name;}
public void setName(String name) {this.name = name;}
//name
}

package project1;
import java.util.ArrayList;

public class Student {
    private int id;
    private String name;
    ArrayList<Book> borrowlist = new ArrayList<Book>();
    private int borrowLimit;
    public Student()
    {

    }

    public void borrowbook(int id, String title, String author, int year, String
publisher)
    { //대출
        Book borrow = new Book(id, title, author, year, publisher);
        System.out.println("책이 정상적으로 대출되었습니다.");
        setBorrowLimit(getBorrowLimit()+1);
    }
}

```

```

        borrowlist.add(borrow);
    }
    public void returnbook(int id, String title, String author, int year, String
publisher)
    { //반납
        System.out.println("책이 정상적으로 반납되었습니다.");
        setBorrowLimit(getBorrowLimit()-1);
        for (int i = 0; i < borrowlist.size(); i++)
        {
            if (borrowlist.get(i).getId() == id &&
borrowlist.get(i).getTitle().equals(title)&& borrowlist.get(i).getAuthor().equals(author)
&& borrowlist.get(i).getYear() == year &&
borrowlist.get(i).getPublisher().equals(publisher))
            {
                borrowlist.remove(i);
                break;
            }
        }
    }
    public int getId() {return id;}
    public void setId(int id) {this.id = id;}
    public String getName() {return name;}
    public void setName(String name) {this.name = name;}
    public int getBorrowLimit() {return borrowLimit;}
    public void setBorrowLimit(int borrowLimit) {this.borrowLimit = borrowLimit;}
}

```

```
package project1;
```

```
public class Book {
```

```

    private int id; //도서의 고유 id
    private int status; //도서의 대출 여부
    private String title; //도서의 제목
    private String author; // 도서의 저자

```

```
private int year; // 도서의 출판연도
private String publisher; //도서의 출판사

public Book (int id, String title, String author, int year, String publisher)
{
    this.setId(id);
    this.setTitle(title);
    this.setAuthor(author);
    this.setStatus(status);
    this.setYear(year);
    this.setPublisher(publisher);
}

public Book ()
{

}

public int getId() {return id;}
public void setId(int id) {this.id = id;}
public int getStatus() {return status;}
public void setStatus(int status) {this.status = status;}
public String getTitle() {return title;}
public void setTitle(String title) {this.title = title;}
public String getAuthor() {return author;}
public void setAuthor(String author) {this.author = author;}
public int getYear() {return year;}
public void setYear(int year) {this.year = year;}
public String getPublisher() {return publisher;}
public void setPublisher(String publisher) {this.publisher = publisher;}
}

package project1;

import java.util.ArrayList;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
```

```

import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
import java.io.FileOutputStream;

public class Library {
    ArrayList<Book> booklist = new ArrayList<Book>(); //모든 도서에 대한 정보
    ArrayList<Admin> adminlist = new ArrayList<Admin>(); //관리자의 목록
    ArrayList<Student> studentlist = new ArrayList<Student>(); //학생의 목록
    private int sn = -1;
    private int an = -1;
    Scanner input = new Scanner(System.in);

    public void createUserDB() throws IOException //user.txt로부터 사용자를
    입력받는 메서드
    {
        String ustr = null;
        String uf = System.getProperty("user.dir") + "\\User.txt";
        BufferedReader ubr = new BufferedReader(new FileReader(uf));
        while ((ustr = ubr.readLine()) != null)
        {
            String udata[] = ustr.split("//");
            if (udata[2].equals("admin"))
            {
                Admin admin = new Admin(Integer.parseInt(udata[0]),
                udata[1]);
                adminlist.add(admin);
            }
            else if (udata[2].equals("student"))
            {
                Student student = new Student();
                student.setId(Integer.parseInt(udata[0]));
                student.setName(udata[1]);
                studentlist.add(student);
            }
        }
    }
}

```

```

    }
    ubr.close();
}

public void createBookDB() throws IOException //book.txt로부터 책을
입력받는 메서드
{
    String bstr = null;
    String bf = System.getProperty("user.dir") + "\\Book.txt";
    BufferedReader bbr = new BufferedReader(new FileReader(bf));
    while ((bstr = bbr.readLine()) != null)
    {
        String bd[] = bstr.split("//");
        Book book = new Book(Integer.parseInt(bd[0]), bd[1], bd[2],
Integer.parseInt(bd[3]), bd[4]);
        booklist.add(book);
    }
    bbr.close();
}

public int searchBook(int id, String title, String author, int year, String
publisher) // 도서의 검색을 담당하는 메서드
{
    for (int i = 0; i < booklist.size(); i++)
    {
        if (booklist.get(i).getId() == id &&
booklist.get(i).getTitle().equals(title) && booklist.get(i).getAuthor().equals(author) &&
booklist.get(i).getYear() == year && booklist.get(i).getPublisher().equals(publisher))
        {
            return i;
        }
    }
    return -1;
}

public int login(int id, String name) //로그인을 담당하는 메서드
{
    for (int i = 0; i < adminlist.size(); i++)

```

```

        {
            if (adminlist.get(i).getId() == id &&
adminlist.get(i).getName().equals(name))
                {
                    an = i;
                    return 0;
                }
        }
        for (int i = 0; i < studentlist.size(); i++)
        {
            if (studentlist.get(i).getId() == id &&
studentlist.get(i).getName().equals(name))
                {
                    sn = i;
                    return 1;
                }
        }
        return -1;
    }
    public void logout() //로그아웃을 담당하는 메서드
    {
        sn = -1;
        an = -1;
        System.out.println("로그아웃되었습니다.");
    }
    public void returnBook(int id, String title, String author, int year, String
publisher) //도서 반납을 담당하는 메서드
    {
        int cf = searchBook(id, title, author, year, publisher);
        int checkbook = -1;
        if (cf != -1) {
            for (int i = 0; i < studentlist.get(sn).borrowlist.size(); i++)
            {
                if (studentlist.get(sn).borrowlist.get(i).getId() == id
&& studentlist.get(sn).borrowlist.get(i).getTitle().equals(title) &&

```

```

studentlist.get(sn).borrowlist.get(i).getAuthor().equals(author) &&
studentlist.get(sn).borrowlist.get(i).getYear() == year &&
studentlist.get(sn).borrowlist.get(i).getPublisher().equals(publisher))
    {
        checkbook = i;
        break;
    }
else
    checkbook = -1;
}
if (checkbook != -1)
{
    if (booklist.get(cf).getStatus() == 1)
    {
        booklist.get(cf).setStatus(0);
        studentlist.get(sn).returnbook(id, title, author,
year, publisher);
    }
}
else
    System.out.println("대출한 도서가 아닙니다.");
}
else
    System.out.println("도서를 찾을 수 없습니다.");
}
public void borrowBook(int id, String title, String author, int year, String
publisher) //도서 대출을 담당하는 메서드
{
    int cf = searchBook(id, title, author, year, publisher);
    if (cf != -1 && studentlist.get(sn).getBorrowLimit() <= 2)
    {
        if (booklist.get(cf).getStatus() == 0)
        {
            booklist.get(cf).setStatus(1);
            studentlist.get(sn).borrowbook(id, title, author, year,

```

```

publisher);
        }
        else
            System.out.println("이미 대출되었습니다.");
    }
    else if (studentlist.get(sn).getBorrowLimit() == 3)
        System.out.println("최대 대출할 수 있는 도서 수는
3권입니다.");
    else
        System.out.println("도서를 찾을 수 없습니다.");
    }
    public void appendBook(int id, String title, String author, int year, String
publisher) //도서의 추가를 담당하는 메서드
    {
        Book append = new Book(id, title, author, year, publisher);
        int cf = -1;
        for (int i = 0; i < booklist.size();i++)
        {
            if (booklist.get(i).getId() == id)
            {
                System.out.println("도서의 ID가 이미 존재합니다.");
                cf = 1;
                break;
            }
        }
        if (cf == -1)
        {
            booklist.add(append);
            System.out.println("도서가 추가되었습니다.");
        }
    }
    public void deleteBook(int id, String title, String author, int year, String
publisher) //도서의 삭제를 담당하는 메서드
    {
        int cf = searchBook(id, title, author, year, publisher);

```

```
        if (cf == -1)
        {
            System.out.println("도서를 찾을 수 없습니다.");
        }
        else
        {
            if (booklist.get(cf).getStatus() == 1)
                System.out.println("도서가 대출중입니다.");
            else {
                booklist.remove(cf);
                System.out.println("도서가 삭제되었습니다.");
            }
        }
    }
}

public Book saveBookDB() // 도서 정보를 저장하는 메서드
{
    System.out.printf("Book ID : ");
    int id = input.nextInt();
    System.out.printf("Book Title : ");
    String title = input.next();
    System.out.printf("Book Author : ");
    String author = input.next();
    System.out.printf("Book Year : ");
    int year = input.nextInt();
    System.out.printf("Book Publisher : ");
    String publisher = input.next();
    Book ifmt = new Book(id, title, author, year, publisher);
    return ifmt;
}

public void modifyBook(int id, String title, String author, int year, String
publisher) //도서의 수정을 담당하는 메서드
{
    int cf = searchBook(id, title, author, year, publisher);
    if (cf == -1)
    {
```

```

        System.out.println("수정을 원하시는 도서의 새 정보를
입력해주세요.");
        booklist.add(saveBookDB());
        System.out.println("도서를 찾을 수 없습니다.");
    }
    else
    {
        if (booklist.get(cf).getStatus() == 1)
        {
            System.out.println("수정을 원하시는 도서의 새
정보를 입력해주세요.");
            booklist.add(saveBookDB());
            System.out.println("도서가 대출중입니다.");
        }
        else
        {
            System.out.println("수정을 원하시는 도서의 새
정보를 입력해주세요.");
            booklist.remove(cf);
            booklist.add(saveBookDB());
            System.out.println("도서가 수정되었습니다.");
        }
    }
}
}
}

```

```
package project1;
```

```
import java.util.Scanner;
import java.io.IOException;
import java.io.IOException;
```

```
public class LibraryTest {
```

```
    public static void main(String args[]) throws IOException
```

```

{
    Library library = new Library();
    int st; //메뉴변수
    String un; //이름 변수
    int uhid; //id변수
    Book book = new Book();
    Scanner input = new Scanner(System.in);
    library.createUserDB();
    library.createBookDB();
    while (true) //메뉴 반복문
    {
        System.out.println("1. 로그인");
        System.out.println("2. 프로그램 종료");
        st = input.nextInt();

        if (st == 2)
        {
            System.out.println("프로그램이 종료되었습니다.");
            break;
        }
        else if (st != 1 && st != 2)
        {
            System.out.println("잘못된 입력입니다. 다시
입력하세요.");
        }
        else
        {
            System.out.printf("ID : ");
            uhid = input.nextInt();
            System.out.printf("NAME : ");
            un = input.next();
            if (library.login(uhid, un) == -1)
            {
                System.out.println("로그인 실패");
                continue;
            }
        }
    }
}

```

```
}
else if (library.login(uhid, un) == 0)
{
    do
    {
        System.out.println("메뉴를 선택해 주세요.");
        System.out.println("1. 도서 추가");
        System.out.println("2. 도서 삭제");
        System.out.println("3. 도서 수정");
        System.out.println("4. 로그아웃");
        System.out.println("5. 프로그램 종료");
        st = input.nextInt();
        switch (st)
        {
            case 1:
                System.out.printf("Book ID : ");
                book.setId(input.nextInt());
                System.out.printf("Book Title : ");
                book.setTitle(input.next());
                System.out.printf("Book Author : ");
                book.setAuthor(input.next());
                System.out.printf("Book Year : ");
                book.setYear(input.nextInt());
                System.out.printf("Book Publisher : ");
                book.setPublisher(input.next());
                library.appendBook(book.getId(),
                book.getTitle(), book.getAuthor(), book.getYear(), book.getPublisher());
                break;
            case 2:
                System.out.printf("Book ID : ");
                book.setId(input.nextInt());
                System.out.printf("Book Title : ");
                book.setTitle(input.next());
                System.out.printf("Book Author : ");
```

```

        book.setAuthor(input.next());
        System.out.printf("Book Year : ");
        book.setYear(input.nextInt());
        System.out.printf("Book Publisher :
");

        book.setPublisher(input.next());
        library.deleteBook(book.getId(),
book.getTitle(), book.getAuthor(), book.getYear(), book.getPublisher());
        break;
    case 3:
        System.out.printf("Book ID : ");
        book.setId(input.nextInt());
        System.out.printf("Book Title : ");
        book.setTitle(input.next());
        System.out.printf("Book Author : ");
        book.setAuthor(input.next());
        System.out.printf("Book Year : ");
        book.setYear(input.nextInt());
        System.out.printf("Book Publisher :
");

        book.setPublisher(input.next());
        library.modifyBook(book.getId(),
book.getTitle(), book.getAuthor(), book.getYear(), book.getPublisher());
        break;
    case 4:
        library.logout();
        break;
    case 5:
        System.out.println(" 프로그램 종료 ");

        System.exit(0);
    default :
        System.out.println(" 잘못된
입력입니다. 다시 입력하세요.");

        break;

```

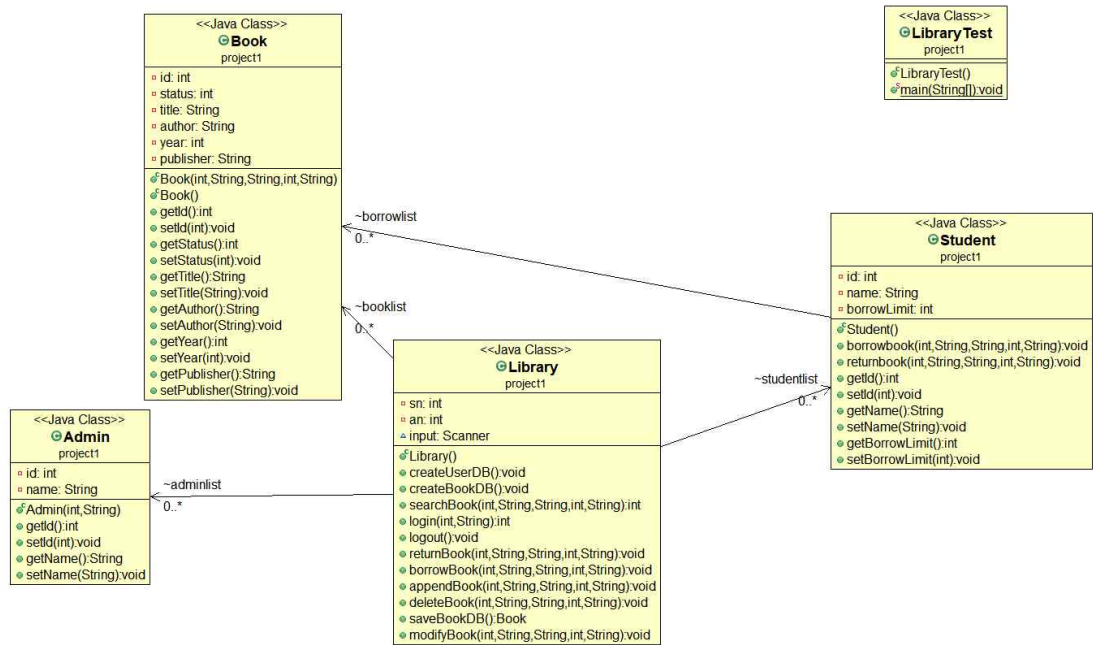
```

    }
} while (st != 4 && st != 5);
}

else if (library.login(uhid, un) == 1)
{
    do {
        System.out.println("메뉴를 선택해 주세요.");
        System.out.println("1. 도서 대출");
        System.out.println("2. 도서 반납");
        System.out.println("3. 로그아웃");
        System.out.println("4. 프로그램 종료");
        st = input.nextInt();
        switch (st)
        {
            case 1:
                System.out.printf("Book ID : ");
                book.setId(input.nextInt());
                System.out.printf("Book Title : ");
                book.setTitle(input.next());
                System.out.printf("Book Author : ");
                book.setAuthor(input.next());
                System.out.printf("Book Year : ");
                book.setYear(input.nextInt());
                System.out.printf("Book Publisher :
");

                book.setPublisher(input.next());
                library.borrowBook(book.getId(),
book.getTitle(), book.getAuthor(), book.getYear(), book.getPublisher());
                break;
            case 2:
                System.out.printf("Book ID : ");
                book.setId(input.nextInt());
                System.out.printf("Book Title : ");
                book.setTitle(input.next());
                System.out.printf("Book Author : ");

```

9주차 - 1

package week09;

```

public class Account {
    private double balance;

    public void deposit(int money) {
        balance += money;
    }

    public void withdraw(int money) throws BalanceException {
        if(balance < money) {
            throw new BalanceException("잔고 부족: " + (money -
            balance) + "원 부족");
        }
        balance -= money;
    }
}

```

package week09;

```
public class AccountTest {  
  
    public static void main(String[] args) {  
        Account account = new Account();  
  
        account.deposit(10000);  
  
        try {  
            account.withdraw(30000);  
        } catch (BalanceException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
package week09;
```

```
public class BalanceException extends Exception {  
    public BalanceException() {  
  
    }  
  
    public BalanceException(String message) {  
        super(message);  
    }  
}
```

```
package week09;
```

```
public class ExceptionExample {  
  
    public static void main(String[] args) {  
        try {
```

```

        throw new NullPointerException();
        // throw new ArrayIndexOutOfBoundsException();
        // throw new Exception();
    } catch(ArrayIndexOutOfBoundsException | NullPointerException e) {
        if(e instanceof ArrayIndexOutOfBoundsException) {
            System.out.println("ArrayIndexOutOfBoundsException
예외가 발생하였습니다.");
            e.printStackTrace();
        }
        else if(e instanceof NullPointerException) {
            System.out.println("NullPointerException 예외가
발생하였습니다.");
            e.printStackTrace();
        }
    } catch(Exception e2) {
        System.out.println("Exception 예외가 발생하였습니다.");
        e2.printStackTrace();
    } finally {
        System.out.println("예외 처리의 종료");
    }
}
}

```

9주차 - 2

```

package hw09;

public class ErrorHandler {
    public static void main(String[] args) {
        int value1 = 0;
        int value2 = 0;
        int result = 0;

        try {
            //프로그램 실행 parameter를 받음

```

```

String input1 = args[0];
String input2 = args[1];

// String type으로 받은 값을 int type으로 변환
value1 = Integer.parseInt(input1);
value2 = Integer.parseInt(input2);

result = value1 + value2;

}
catch(ArrayIndexOutOfBoundsException e1) {
    System.out.println("ArrayIndexOutOfBoundsException 예외가
발생하였습니다.");
    e1.printStackTrace();
}
catch(NumberFormatException e2) {
    System.out.println("NumberFormatException 예외가
발생하였습니다.");
    e2.printStackTrace();
}
finally {
    System.out.println("결과는 " + result + " 입니다.");
}

}
} ‘

```

10주차 - 1

```

module JavaFX {
    requires javafx.controls;
    requires javafx.fxml;

    opens application to javafx.graphics, javafx.fxml;
}

```

```
}

package application;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

//ctrl + shift + o

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            Parent root =
FXMLLoader.load(getClass().getResource("MyApp.fxml"));
            Scene scene = new Scene(root);

            primaryStage.setTitle("week10");
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}

<?xml version="1.0" encoding="UTF-8"?>
```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane xmlns:fx="http://javafx.com/fxml/1"
xmlns="http://javafx.com/javafx/16">
  <children>
    <AnchorPane prefHeight="150.0" prefWidth="350.0">
      <children>
        <Button layoutX="150.0" layoutY="100.0" mnemonicParsing="false"
text="Button" />
      </children>
    </AnchorPane>
    <Label layoutX="35.0" layoutY="10.0" text="Hello, JavaFX">
      <font>
        <Font size="48.0" />
      </font>
    </Label>
  </children>
</AnchorPane>

```

11주차 - 1

```
package temp;
```

```

import java.beans.XMLDecoder;
import java.io.BufferedOutputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.util.ArrayList;

```

```

public class SerializeToXML {
    private static final String SERIALIZED_FILE_NAME = "dvd2.xml";

```

```
public static void main(String args[]) {

    Movie bourneIdentity = new Movie("The Bourne Identity", 119,
    "Doug Liman", 2002, "Matt Damon, Franka Potente");
    Movie bourneSupermacy = new Movie("The Bourne Supermacy",
    108, "Paul Greengrass", 2004, "Matt Damon, Franka Potente, Joan Allen");
    Movie bourneUltimatum = new Movie("The Bourne Ultimatum", 115,
    "Paul Greengrass", 2007, "Matt Damon, Edgar Ramirez, Joan Allen");
    Movie bourneLegacy = new Movie("The Bourne Legacy", 135,
    "Tony Gilroy", 2012, "Jeremy Renner, Rachel Weisz, Edward Norton");

    ArrayList<Movie> moviesList = new ArrayList<>();
    moviesList.add(bourneIdentity);
    moviesList.add(bourneSupermacy);
    moviesList.add(bourneUltimatum);
    moviesList.add(bourneLegacy);

    DVD bourneSeries = new DVD();
    bourneSeries.setMovies(moviesList);

    XMLEncoder encoder = null;
    try
    {
        encoder = new XMLEncoder(new BufferedOutputStream(new
    FileOutputStream(SERIALIZED_FILE_NAME)));
        System.out.println("정상변환");
    }
    catch(FileNotFoundException fileNotFound)
    {
        System.out.println("ERROR: While Creating or Opening the
    File dvd.xml");
    }
    encoder.writeObject(bourneSeries);
    encoder.close();
}
```

```
    }  
}  
  
package temp;  
  
public class Movie {  
    private String name;  
    private int runtime;  
    private String directors;  
    private int released;  
    private String cast;  
  
    public Movie() {  
  
    }  
  
    public Movie(String name, int runtime, String directors, int released, String  
cast) {  
        this.name = name;  
        this.runtime = runtime;  
        this.directors=directors;  
        this.released=released;  
        this.cast=cast;  
    }  
  
    public String getCast() {  
        return cast;  
    }  
  
    public void setCast(String cast) {  
        this.cast = cast;  
    }  
  
    public int getReleased() {  
        return released;  
    }  
}
```

```
    }

    public void setReleased(int released) {
        this.released = released;
    }

    public String getDirectors() {
        return directors;
    }

    public void setDirectors(String directors) {
        this.directors = directors;
    }

    public int getRuntime() {
        return runtime;
    }

    public void setRuntime(int runtime) {
        this.runtime = runtime;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

package temp;

import java.util.ArrayList;
```

```

public class DVD {
    private ArrayList<Movie> movies = new ArrayList<>();
    public DVD() {

    }
    public ArrayList<Movie> getMovies() {
        return movies;
    }

    public void setMovies(ArrayList<Movie> movies) {
        this.movies = movies;
    }

    public String toString() {
        String movies = "";
        for(Movie movie:getMovies()) {
            movies += "영화명: " + movie.getName() + ", 러닝타임: "
+ movie.getRuntime() + ", 제작: " + movie.getDirectors() + ", 상영년도: " +
movie.getReleased() + ", 주요 출연진: " + movie.getCast() + "\n";
        }
        return movies;
    }
}

```

```
package temp;
```

```

import java.beans.XMLDecoder;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileInputStream;

```

```

public class DeserializeFromXML {
    private static final String SERIALIZED_FILE_NAME = "dvd2.xml";

```

```
public static void main(String[] args) {
    XMLDecoder decoder=null;

    System.out.println("[MARVEL Cinematic Universe] print by
202021167 하정민");

    try
    {
        decoder=new XMLDecoder(new BufferedInputStream(new
FileInputStream(SERIALIZED_FILE_NAME)));
    }

    catch (FileNotFoundException e)
    {
        System.out.println("ERROR: File dvd2.xml not found");
    }

    DVD bourneSeries=(DVD)decoder.readObject();
    System.out.println(bourneSeries);
}
}
```

12주차 - 1

```
package week12;
```

```
import java.util.Objects;
```

```
public class Student {
    private int stuNum;
    private String name;

    public Student(int stuNum, String name) {
        this.stuNum = stuNum;
        this.name = name;
    }
}
```

```
}

public String toString() {
    return String.format("%d :: %s", stuNum, name);
}

public int hashCode() { /* 동일 학번이면 정수값이 return 되도록 작성 */
    return Objects.hash(stuNum);
}

public boolean equals(Object obj) {
    if(obj instanceof Student) {
        // 내용 작성
        Student student = (Student) obj;

        return (stuNum == student.stuNum);
    }
    else {
        return false;
    }
}

private int getStuNum() {
    return stuNum;
}

private void setStuNum(int stuNum) {
    this.stuNum = stuNum;
}

private String getName() {
    return name;
}

private void setName(String name) {
```

```
        this.name = name;
    }
}

package week12;

import java.util.List;
import java.util.*;

public class PointTest {

    public static void main(String[] args) {

        List<Point> pointList = new LinkedList<>();

        MyComparator myComparator = new MyComparator();

        pointList.add(new Point(2, 3));
        pointList.add(new Point(4, 6));
        pointList.add(new Point(7, 2));
        pointList.add(new Point(5, 5));

        System.out.println("before sorting");
        //List 출력 코드
        System.out.println(pointList);

        // 정렬 코드
        Collections.sort(pointList, myComparator);

        System.out.println("after sorting");
        // List 출력 코드
        //System.out.println(pointList);

        Iterator<Point> iter = pointList.iterator();
        while(iter.hasNext()) {
```

```
        System.out.println(iter.next());
    }
}
}
```

```
package week12;
```

```
public class Point {
    private int x;
    private int y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public String toString() {
        return String.format("x: %d, y: %d", x, y);
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }
}
```

```
package week12;

import java.util.Comparator;

public class MyComparator implements Comparator<Point>{

    public int compare(Point o1, Point o2) {
        if(o1.getY() < o2.getY()) {
            return 1;
        }
        else {
            return -1;
        }
    }
    // Collections.sort() -> compare(자료구조, 정렬기준) }
}
```

```
package week12;

import java.util.HashSet;
import java.util.Iterator;

public class HandSetTest {

    public static void main(String[] args) {
        HashSet<Student> set = new HashSet<>();

        set.add(new Student(202012345, "김국디"));
        set.add(new Student(202098765, "이아주"));
        set.add(new Student(202012345, "박정통"));

        //Set의 전체 원소들을 출력 코드 작성

        Iterator<Student> iter = set.iterator();
        while(iter.hasNext()) {
            System.out.println(iter.next());
        }
    }
}
```

```
        }  
    }  
}
```

12주차 - 2

```
package hw12;
```

```
import java.io.*;  
import java.util.*;
```

```
public class ProcessData {  
    private int eng;  
    private int mat;  
    private int kor;  
    private int stu;
```

```
    ArrayList<ProcessData> stulist = new ArrayList<ProcessData>();  
    MyComparator myComparator = new MyComparator();
```

```
    public ProcessData() {  
  
    }  
}
```

```
    public ProcessData(int stu, int kor, int mat, int eng) {  
        this.stu = stu;  
        this.kor = kor;  
        this.mat = mat;  
        this.eng = eng;  
    }  
}
```

```
    public void readData(String name) throws IOException{
```

```
        BufferedReader in = new BufferedReader(new
```

```

FileReader(name));

        String s;
        while ((s = in.readLine()) != null)
        {

                String[] split = s.split(",");

                stulist.add(new ProcessData(stu =
Integer.valueOf(split[0]), kor = Integer.valueOf(split[1]), mat =
Integer.valueOf(split[2]), eng = Integer.valueOf(split[3]));
        }
        in.close();
    }

    public void sortData() {
        Collections.sort(stulist, myComparator);
    }
    public void printData() {
        for(ProcessData a: stulist)
            System.out.println("학번: " + a.getStu() + ", " + "국어: " +
a.getKor() + ", " + "수학: " + a.getMat() + ", " + "영어: " + a.getEng());
    }

    public int getEng() {return eng;}
    public void setEng(int eng) {this.eng = eng;}
    public int getMat() {return mat;}
    public void setMat(int mat) {this.mat = mat;}
    public int getKor() {return kor;}
    public void setKor(int kor) {this.kor = kor;}
    public int getStu() {return stu;}
    public void setStu(int stu) {this.stu = stu;}
}

```

```
package hw12;
```

```
import java.util.*;

public class MyComparator implements Comparator<ProcessData>{

    public int compare(ProcessData a, ProcessData b) {
        if(a.getEng() < b.getEng())
            return 1;
        else if(a.getEng() == b.getEng())
        {
            if(a.getMat() < b.getMat())
                return 1;
            else if(a.getMat() == b.getMat())
            {
                if(a.getKor() < b.getKor())
                    return 1;
                else if(a.getKor() == b.getKor())
                {
                    if(a.getStu() < b.getStu())
                    {
                        return -1;
                    }
                    else
                        return -1;
                }
                else
                    return -1;
            }
        }
        else {
            return -1;
        }
    }
}
```

```
    }  
}  
  
package hw12;  
  
import java.io.IOException;  
  
public class Main {  
    public static void main(String[] args) {  
        ProcessData process = new ProcessData();  
        String fname = "C:/Users/hadil/OneDrive/바탕 화면/2학년  
1학기/이클립스/202021167_하정민_HW12/input.txt";  
        try {  
            System.out.println("input Data");  
            process.readData(fname);  
            System.out.println("sorting..");  
            process.sortData();  
            System.out.println("after sorting");  
            process.printData();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

13주차 - 1

```
package hw13;  
  
import java.io.File;  
import java.io.FileWriter;
```

```
import java.io.IOException;

public class WriteText<T, M> extends IO_Class<T, M> {

    private T input;
    private M output;

    public WriteText(T input, M output) {
        super(input, output);
    }

    public void writeData(M output, String to, ReadText rt) {
        this.output = output;

        if(this.output instanceof String) {
            try
            {
                File file = new File(to);
                FileWriter filewriter;

                filewriter = new FileWriter(file, false);
                filewriter.write(rt.getTextString());
                System.out.println("출력 완료");

                filewriter.flush();
                filewriter.close();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        else
        {
            System.out.println("알 수 없는 데이터 타입입니다.");
        }
    }
}
```

```
    }  
  }  
}  
  
package hw13;  
  
import java.io.FileReader;  
import java.io.IOException;  
  
public class ReadText<T> {  
  
    private String textString = null;  
    private int numberOfSpace = 0;  
    private int alphanumber = 4;  
    private T input;  
  
    public ReadText(T input) {  
        this.input = input;  
    }  
  
    void readData(T input, String from) {  
        this.input = input;  
  
        int textinout = 0;  
  
        if(this.input instanceof String) {  
            try {  
                FileReader filereader = new FileReader(from);  
  
                while(true) {  
                    char[] tentext = new char[10];  
  
                    textinout = filereader.read(tentext, 0, 10);  
  
                    if(textinout == -1)
```

```
        {
            break;
        }

        for(char text : tentext)
        {
            if('a' <= text && text <= 'z')
            {
                textString = textString + text;
            }
            else
            {
                numberOfSpace++;
            }
            alphanumber++;
        }
    }

    setAlphanumber(alphanumber);
    setNumberOfSpace(numberOfSpace);
    filereader.close();

} catch (IOException e) {
    e.printStackTrace();
}
} else {
    System.out.println("알 수 없는 데이터 타입입니다.");
}
}

public String getTextString() {
    return textString;
}

public void setTextString(String textString) {
    this.textString = textString;
}
```

```
}  
public int getNumberOfSpace() {  
    return numberOfSpace;  
}  
public void setNumberOfSpace(int numberOfSpace) {  
    this.numberOfSpace = numberOfSpace;  
}  
public T getInput() {  
    return input;  
}  
public void setInput(T input) {  
    this.input = input;  
}  
public int getAlphanumeric() {  
    return alphanumeric;  
}  
public void setAlphanumeric(int alphanumeric) {  
    this.alphanumeric = alphanumeric;  
}  
}
```

```
package hw13;
```

```
public class Main {
```

```
    public static void main (String[] args) {
```

```
        String from = "C:\\Users\\hadil\\OneDrive\\바탕 화면\\2학년  
1학기\\이클립스\\202021167_하정민_HW13\\src\\input.txt";
```

```
        String to = "C:\\Users\\hadil\\OneDrive\\바탕 화면\\2학년  
1학기\\이클립스\\202021167_하정민_HW13\\src\\output.txt";
```

```
        String input = "input";
```

```
        String output = "output";
```

```
ReadText<String> readtext = new ReadText<>(input);
WriteText<String, String> writetext = new WriteText<>(input, output);

readtext.readData(input, from);
System.out.println("text 파일의 총 글자 수: " +
readtext.getAlphanumeric());
System.out.println("text 파일의 알파벳 수: " +
(readtext.getAlphanumeric() - readtext.getNumberOfSpace()));
writetext.writeData(output, to, readtext);
}
}
```

```
package hw13;
```

```
public class IO_Class<T, M> {

    private T input;
    private M output;

    public IO_Class(T input, M output) {
        this.input = input;
        this.output = output;
    }

    public T getInput() {
        return input;
    }

    public void setInput(T input) {
        this.input = input;
    }

    public M getOutput() {
        return output;
    }

    public void setOutput(M output) {
        this.output = output;
    }
}
```

```
    }
}
```

14주차 - 1

```
package week14;
```

```
public class MathTest {
    public static void main(String[] args) {
        Math plusLambda = (first, second) -> first + second;
        System.out.println(plusLambda.Calc(4, 2));

        Math minusLambda = (first, second) -> first - second;
        System.out.println(minusLambda.Calc(4, 2));

        Math multiLambda = (first, second) -> first * second;
        System.out.println(multiLambda.Calc(4, 2));

        Math divideLambda = (first, second) -> first / second;
        System.out.println(divideLambda.Calc(4, 2));
    }
}
```

```
package week14;
```

```
public interface Math {
    public int Calc(int first, int second);
}
```

```
package week14;
```

```
import java.util.List;
import java.util.stream.Collectors;
```

```
public class Intermediate {
```

```

public static void main(String[] args) {
    List<String> list = List.of("a", "ab", "abc", "abcd");

    List<String> result = list.stream()
        .filter(x -> { //중간 연산 1
            System.out.println(x + " in filter method");
            return x.length() >= 1;
        }).map(x -> { //중간 연산 2
            System.out.println(x + " in map method");
            return x.toUpperCase();
        }).limit(2) // 중간 연산 3
        .collect(Collectors.toList()); // 나중에 살펴볼 단말

```

연산

```

        System.out.println(result);
        //a in filter method
        //a in map method
        //ab in filter method
        //ab in map method
        // [A, AB]
    }
}

```

14주차 - 2

```

package hw14;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class Main2 {

```

```

public static void main(String[] args) {
    List<String> earthlist = new ArrayList();
    try {
        BufferedReader in = new BufferedReader(new
FileReader("C:/Users/dillo/Desktop/2학년
1학기/EST/202021167_하정민_HW14/input.txt"));
        String a;
        while ((a = in.readLine()) != null) {
            String[] split = a.split(" ");
            for(int i = 0; i<split.length; i++) {
                earthlist.add(split[i]);
            }
        }
        in.close();
    } catch (IOException e) {
        System.err.println(e);
    }
    List<String> result = earthlist.stream().filter(x -> {return
x.contains("al");}).map(x -> {return x.toUpperCase();}).collect(Collectors.toList());
    System.out.println(result);
}
}

```

```
package hw14;
```

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

```

```

public class Main {
    public static void main(String[] args) {
        List<String> earthlist = new ArrayList();

```

```

        try {
            BufferedReader in = new BufferedReader(new
FileReader("C:/Users/dillo/Desktop/2학년
1학기/EST/202021167_하정민_HW14/input.txt"));
            String a;
            while ((a = in.readLine()) != null) {
                String[] split = a.split(" ");
                for(int i = 0; i<split.length; i++) {
                    earthlist.add(split[i]);
                }
            }
            in.close();
        } catch (IOException e) {
            System.err.println(e);
        }
        List<String> result = earthlist.stream().filter(x -> {return
x.equals("earth's");}).map(x -> {return x.toUpperCase();}).collect(Collectors.toList());
        System.out.println(result);
    }
}

```

15주차 - 1

package week15;

```

public class BeepPrintExample1 {
    public static void main(String[] args) {

        Runnable beepTask = new BeepTask();
        Thread thread1 = new Thread(beepTask);
        thread1.start();

        for(int i = 0; i < 5 ; i++) {
            System.out.println("뽕");
            try {Thread.sleep(500);} catch(Exception e) {}
        }
    }
}

```

```
    }  
  }  
}
```

```
package week15;
```

```
import java.awt.Toolkit;
```

```
public class BeepThread extends Thread {  
    public void run() {  
        Toolkit toolkit = Toolkit.getDefaultToolkit();  
        for(int i = 0; i < 5 ; i++) {  
            toolkit.beep();  
            try {Thread.sleep(500);} catch(Exception e) {}  
        }  
    }  
}
```

```
package week15;
```

```
import java.awt.Toolkit;
```

```
public class BeepTask implements Runnable {  
    public void run() {  
        Toolkit toolkit = Toolkit.getDefaultToolkit();  
        for(int i = 0; i < 5 ; i++) {  
            toolkit.beep();  
            try {Thread.sleep(500);} catch(Exception e) {}  
        }  
    }  
}
```

```
package week15;
```

```
public class BeepPrintExample3 {
    public static void main(String[] args) {
        Thread thread = new BeepThread();
        thread.start();

        for(int i = 0; i < 5 ; i++) {
            System.out.println("땡");
            try {Thread.sleep(500);} catch(Exception e) {}
        }
    }
}

package week15;

import java.awt.Toolkit;

public class BeepPrintExample2 {
    public static void main(String[] args) {

        Thread thread2 = new Thread() -> {
            Toolkit toolkit = Toolkit.getDefaultToolkit();
            for(int i = 0; i < 5 ; i++) {
                toolkit.beep();
                try {Thread.sleep(500);} catch(Exception e) {}
            }
        };
        thread2.start();

        for(int i = 0; i < 5 ; i++) {
            System.out.println("땡");
            try {Thread.sleep(500);} catch(Exception e) {}
        }
    }
}
```

15주차 - 2

```
package hw15;

public class User2 extends Thread implements Runnable {
    private Calculator calculator;

    public void setCalculator(Calculator calculator) {
        this.setName("User2");
        this.calculator = calculator;
    }

    public void run() {
        calculator.setMemory(50);
    }
}
```

```
package hw15;

public class User1 extends Thread {
    private Calculator calculator;
    Thread thread1 = Thread.currentThread();

    public User1() {

    }

    public void setCalculator(Calculator calculator) {
        this.setName("User1");
        this.calculator=calculator;
    }

    public void run() {
        calculator.setMemory(100);
    }
}
```

```
}
```

```
package hw15;
```

```
public class MainThread {  
    public static void main(String[] args) {  
  
        Calculator calculator = new Calculator();  
  
        User1 user1 = new User1();  
        User2 user2 = new User2();  
  
        user1.setCalculator(calculator);  
        user2.setCalculator(calculator);  
  
        user1.start();  
        user2.start();  
    }  
}
```

```
package hw15;
```

```
public class Calculator {  
    private int memory;  
  
    public int getMemory() {  
        return memory;  
    }  
  
    public synchronized void setMemory(int memory) {  
        this.memory = memory;  
    }  
}
```

```
System.out.print(Thread.currentThread().getName() + ": ");
```

```
System.out.println(getMemory());
```

```
try
```

```
{
```

```
    Thread.sleep(2000);
```

```
}
```

```
catch (InterruptedException e)
```

```
{
```

```
}
```

```
}
```

```
}
```

입선

비대면으로 소통하는 법

교과목명 : 전자회로1

교수명 : 오일권

학과 : 전자공학과

이름 : 김기훈

1. 수업 참여 후기

이번 학기 역시 코로나로 인해 대부분의 수업이 비대면으로 진행되었습니다. 비대면 수업은 일반적으로 녹화된 영상을 듣고, 온라인으로 과제를 제출하는 방식입니다. 그래서 대부분의 수업은 교수님이나 학우들과 소통할 기회가 없습니다. 그러나 오일권 교수님의 전자회로1 수업은 다른 비대면 수업과 달리 아주bb로 소통을 많이 해 인상적이었습니다.

수업은 기본적으로 동영상 강의였습니다. 월, 수 A교시였는데, 교수님께선 8시 30분에 강의를 올려주시고 10시 15분까지 강의를 들어야만 출석으로 인정하셨습니다. 이렇게 실제 대면 수업처럼 시간을 맞춰야 하는 방식은 학생들이 부지런하게 강의를 듣고, 밀리지 않게 만들었습니다. 시험 전후에는 실시간 zoom으로 학생들과 소통하셨습니다. 시험 전에는 중요한 부분을 같이 복습하고, 회로에 대한 사고방식을 알려주며 공부하는 방향을 잡아주셨습니다. 혼자 공부하면 방향을 잘못 잡을 수 있는데, 교수님이 잘 이끌어 주셔서 큰 도움이 되었습니다. 시험 후에는 학생들의 진로에 대해 상담해주셨습니다. zoom의 소그룹 회의 기능을 이용해 같은 진로를 희망하는 학생들끼리 서로 이야기하고, 궁금한 점을 모아 질문하도록 하셨습니다. 비대면 수업에선 같은 수업을 듣는 학우들을 한 번도 못 보는 경우가 대부분인데, 얼굴을 보며 진로에 관한 대화를 나누니 학교 다니는 느낌이 나서 좋았습니다. 또한 반도체의 역사와 공정에 대한 영상도 틈틈이 보여주시며 학생들의 진로 선택에 도움을 주셨습니다.

과제는 주 단위로 주셨습니다. 그리고 교수님께서 우수한 과제를 선별해 그 학생에게는 free pass권을 주셨습니다. free pass권은 자신이 원할 때 한 주 과제를 pass할 수 있는 것입니다. 교수님께선 이렇게 학생들이 과제를 열심히 하게끔 독려하셨

습니다. 저도 freepass권을 받은 적이 있는데, 교수님께 인정받으니 기분이 좋았고, 이 과목에 흥미가 생겨 더 열심히 하게 되었습니다. 비대면 수업에서는 교수님께 칭찬받을 일이 거의 없는데, 이렇게 과제를 잘하면 칭찬받을 수 있으니 좋은 제도라고 생각합니다. 교수님께선 아주bb의 q&a 게시판도 적극적으로 활용하셨습니다. q&a 게시판에 질문을 올리거나, 다른 학우가 올린 질문에 맞는 답을 하면 가산점을 주셨습니다. 그리고 우수한 질문과 답변은 강의에서 언급하며 칭찬하셨습니다. 또한 아주bb의 블로그 기능도 이용하셨습니다. 블로그에서는 모두가 글을 쓰거나 읽을 수 있는데, 교수님께선 시험 전에 개인당 3문제씩 직접 문제를 만들어 올리도록 하셨습니다. 그리고 중간, 기말고사의 50%를 여기서 내셨습니다. 시험문제의 반이 여기서 나오니, 학생들은 모르는 것이 있거나 이상한 부분이 있으면 바로 코멘트를 남겨 소통했습니다. 이렇게 전자회로1 수업에선 학생들이 q&a게시판과 블로그에서 활발하게 소통했습니다. 소통하며 얻은 지식은 더 빨리 이해되고, 기억에 더 오래 남았습니다.

시험은 실시간 zoom으로 진행되었습니다. 전자회로1 과목은 공식이 매우 많습니다. 교수님께선 학생들이 공식을 외우는 데 매몰되지 않고, 회로 설계하는 능력을 키우게 하기 위해 시험에서 A4 한 장의 요약 종이를 허용하셨습니다. 덕분에 단순 암기가 아닌, 회로에 대한 이해와 회로를 구성하는 사고력을 이 수업에서 얻을 수 있었습니다.

2. 나만의 학습 노하우

1. 수업 두 번 듣기

한 시간 가량의 수업을 듣다 보면 분명 집중이 흐트러지고, 놓치는 부분이 생깁니다. 그래서 저는 수업을 두 번 듣습니다. 처음에는 필기하지 않고 흐름과 맥락을 생각하며 듣습니다. 그 후 비대면 강의의 가장 큰 장점인, 다시 들을 때 필기합니다. 이때 정지 기능을 이용해 놓치는 부분 없이 필기합니다. 이렇게 수업을 2번 들으면 내가 놓쳤던 부분을 채우고, 처음 들을 때는 몰랐던 교수님의 의도를 파악할 수 있습니다. 필기할 때는 교수님께서 강조하는 부분과 수업 자료에 없는 부분을 집중적으로 필기합니다. 강조하는 부분과 수업 자료에 없는 내용이 시험에서 변별력 있는 문제로 나오기 때문입니다. 그 후 필기 내용을 다시 한번 정리합니다. 필기만 하는 것은 그냥 받아쓰기입니다. 필기 내용을 바탕으로 스스로 설명하는 과정이 공부입니다.

2. 설명하기

필기를 끝냈다면, 이제 수업 자료와 필기 내용을 스스로 자신에게 설명할 차례입니다. 수업을 들을 때는 이해했다고 생각하지만, 그 부분을 혼자 설명하려 하면 어려울 것입니다. 그 이유는 아직 완전히 이해하지 못했기 때문입니다. 즉, 설명할 수 없으면 아직 모르는 것입니다. 설명할 때는 개념부터 출발해 결론까지의 과정이 논리적이어야 합니다. 과정마다 자신에게 ‘왜?’ 라는 질문을 하고, 이에 답할 수 있어야 합니다. 이렇게 설명하기 위해선 본질적인 정의까지 알아야 하기 때문에, 그동안 배운 것들을 다시 복습해야 합니다. 따라서 초반에는 시간이 되게 많이 걸립니다. 1학년 때 배운 과목의 내용까지 내려가서 공부해야 하기 때문입니다. 하지만 한 번 기초를 잘 공부하면, 이를 바탕으로 공부속도는 더 빨라집니다. 모래성을 쌓을 때, 빨리 쌓기 위해 밑을 얇게 만들고 쌓으면 결국 무너집니다. 시간이 걸리더라도 밑을 두껍게 만들어야, 위로 계속 쌓더라도 무너지지 않습니다. 이 공부방식은 머리에 오래 남고, 면접 준비도 되기 때문에 학년이 올라갈수록 빛을 받습니다.

3. 문제 풀이

그 후 과제를 통해 개념을 적용합니다. 비대면 강의는 과제가 많아서, 평소 과제를 바로 끝내야 시험 기간에 시험공부를 할 수 있습니다. 과제를 할 때는 위에서 언급한 설명하기를 하며 풀입니다. 하지만 시험 준비는 다릅니다. 정확하고 빠르게 문제 푸는 연습이 시험 기간에 필요합니다. 또한 문제를 풀며 자주 실수하는 부분은 노트에 따로 정리합니다. 한번 실수한 부분에서 다시 실수할 확률이 높으므로 실수하는 부분은 항상 의식해야 합니다. 그리고 시험환경을 미리 구성해 공부해보면 보완할 점(자리 배치, 조명 등)을 찾을 수 있고, 실제 시험에서 좀 더 편해집니다.

스스로 설명하다 막히거나, 문제를 풀다 이해되지 않는 내용이 생길 수 있습니다. 그럴 때는 먼저 인터넷으로 찾아보고, 주위에 물어보는 등 알기 위해 노력해야 합니다. 하루 정도 고민해도 모르겠다면, 그때 교수님께 메일로 질문해야 합니다. 비대면 수업이다 보니 대부분 교수님들께선 메일로 질문하는 것을 환영하십니다. 질문을 통해 학생들은 모르는 것을 알 수 있고, 교수님께서도 학생들의 이해도를 파악하고 수업 방향을 정할 수 있습니다. 그러니 교수님과 많이 소통해야 합니다.

4. 분위기 조성

비대면 수업은 집중력 싸움입니다. 기숙사나 집에서 공부하면 아무래도 긴장이

풀리고, 집중력이 떨어집니다. 그래서 스스로 공부하는 분위기를 만들어야 합니다. 우선 강의는 강의 시간에 맞게 들어야 합니다. 그래야 강의를 밀리지 않고, 강의와 과제가 얼마나 남아있는지 헛갈리지 않습니다. 또한 같은 화면을 오랜 시간 동안 보면 집중도가 떨어지기 때문에 강의를 몰아서 듣는 것은 피해야 합니다. 대면 수업에서 강의실 옮기는 것처럼 도서관, 카페 등 주기적으로 장소를 옮겨가며 수업을 들으면 집중력을 올릴 수 있습니다. 학우들과 교류도 중요합니다. 혼자 공부하면 자신이 잘못 이해한 부분을 인식하기 어렵습니다. 이는 자기도 모르게 오답을 유발합니다. 그래서 아주bb나 학교 커뮤니티를 이용해 학우들과 소통하며 내가 이해한 개념이 맞는지 확인해야 합니다.

비대면 수업의 핵심은 소통입니다. 언제 어디서나 강의를 들을 수 있기에 지식을 습득하는 환경은 대면 수업보다 더 좋아졌습니다. 그러나 직접 만날 수 없기에, 배운 지식을 소통하는 환경은 어려워졌습니다. 그래서 온라인으로 소통하는 법에 익숙해져야 합니다. 비대면 수업에서 소통을 활발히 한다면 대면 수업보다 더 좋은 학습 성과를 얻을 수 있습니다.

입선

나를 찾는 여행

교과목명 : 예술이란무엇인가

교 수 명 : 안지연

학 과 : 금융공학과

이 름 : 김유미

1. 수업 참여 후기

수업에서 다뤘던 여러 강의와 과제 중에서 가장 기억에 남았던 과제는 단연 ‘DREAM’이라는 영화를 보고 감상을 제출하는 것이었습니다. 그 주의 강의내용은 빈센트 반고흐의 그림들을 배우는 수업이었고, 저는 이 강의를 아직도 잊지 못할 정도로 생생히 기억합니다. 왜냐하면, 작년에 들었던 ‘영어2’ 수업에서 반고흐의 생전 마지막 편지를 읽으며 매우 감동했던 경험이 있는데 실제 그의 작품을 강의를 통해 전시회처럼 느끼고 더 나아가 영화로 그를 만나니 훨씬 더 심층적인 감정을 유도했습니다. 해당 강의를 하나의 내용을 영어로만, 예술로만 바라보는 것이 아니라 그것들을 동시에 바라보는 법을 저에게 알려주었고 두 강의를 넘나들며 수업을 듣게 했던 뜻깊은 경험으로 남아 있습니다. 고등학교 시절 사회문화 시간에 ‘간학문적 연구’라는 하나의 주제를 여러 학문으로 바라보는 탐구법을 배운 적이 있는데 강의를 들으면서 간학문적 연구와 매우 밀접한 방식으로 수업을 듣고 있다는 생각이 계속해서 머리를 맴돌았습니다. 실제 대학에 와서 이론으로만 배워왔던 새로운 접근법을 배우게 된 것 같아서 신기했고, 이때의 경험은 그 무엇보다도 바꿀 수 없는 진귀한 경험이라고 느꼈습니다.

‘예술이란 무엇인가’ 수업을 저의 언어로 간단히 정리하자면 ‘나를 찾는 여행’입니다. 매주 부여되는 과제가 쉽지만은 않은 과정이었다고 기억합니다. 하지만 강의가 열려 있는 느낌으로 진행되어 자유롭게 저의 의견을 내비칠 수 있었고, 강의 후반부로 갈수록 과제 안에서 나라는 존재에 대한 깊이 있는 탐구로 이어졌습니다. ‘내가 지금 걱정하고 있는 것이 무엇인지’ 과제를 통해 고민하고 고민하여 과제의 답을 제출하였습니다. 해당 과제는 제가 과제를 해결하는 과정 또한 의미 있었지만, 교수님의 답변에 많은 힘을 얻었습니다. 블랙보드의 과제 제출 기능을 통해 비록 비대면 수업이었지만 교수님의 한마디는 저에게 큰 감동이 되어 다가왔습

니다.

또한 이 수업에서 저의 새로운 관심사를 처음 깨닫게 되었습니다. 예술 작품으로 만들어보고 싶은 것 하나를 적는 과제에서 저는 오랜 고민 끝에 우주 공간을 거의 현실과 같은 모습으로 시현해보고 싶다는 대답을 내놓게 되었고, 이 과제는 저를 또 다른 SF 도서나 동영상 등의 정보 매체로 이끌었습니다. ‘우리에게 그 어떠한 제약도 없다면 어떤 예술 작품을 만들 것인가?’ 지금까지 한 번도 생각해본 적 없는 질문이었고 결국 저는 이 답을 통해 새로운 학문에 대해 첫걸음을 내딛게 되었습니다.

COVID-19 이후 온라인(비대면) 강의로 수업을 듣게 되면서 사람과의 소통이 줄어들고 전공 공부에 힘들어했던 시기에 저는 과제의 답을 찾으면서 저를 다시 한번 돌아보았습니다. 여러 가지로 힘들었던 때 이 강의를 듣게 된 건 저에게 크나큰 행운이었다고 생각합니다. 강의를 듣던 시간, 과제를 고민하던 시간, 과제를 제출하던 시간 어느 하나 버릴 것 없이 소중한 시간이었으며 마음속 깊이 간직하고 싶은 강의라고 자부할 수 있습니다.

2. 나만의 학습 노하우

매일 아침을 깨우는 느낌으로 교수님의 강의를 일어나자마자 듣고, 매주 수요일의 과제 제출을 위해 3일 정도 과제의 답을 브레인스토밍하는 방식으로 답에 접근하였습니다. 그리고 수요일 아침에는 그동안 머릿속에서 떠올린 여러 생각을 정리하여 하나의 과제로 제출하였습니다. 교수님의 피드백을 토대로 다음 과제에는 잘못된 부분을 조금씩 해결하려고 노력하였습니다. 실제로 강의 전반부에는 부정확한 접근으로 조금씩 점수 감점이 있긴 했지만, 후반부로 갈수록 정확한 답에 가까워져서 좋은 결과를 이끌어냈던 것 같습니다.

제가 생각하는 비대면 강의의 가장 큰 장점은 일시 정지 기능입니다. 강의에서 나온 작품이 마음에 와닿으면 일시 정지를 하고 해당 예술가의 다른 작품이나 명언 같은 것을 확인해보는 식으로 교수님이 만들어주신 강의노트 옆에 작은 메모로 저만의 새로운 강의노트를 만들어 나갔습니다. 그렇게 차곡차곡 쌓인 강의노트를 바라보면서 뿌듯함을 느끼기도 했고 여름방학에 강의에서 나왔던 도서나 영화를 봐야겠다고 다짐하기도 했습니다. 이런 식의 공부방식은 강의를 제게 단순히 지식을 전달해주는 매개체를 넘어 강의 속에 제가 존재할 수 있도록 만들어주었습니다. 비록 비대면 수업으로 진행되었지만 충분히 강의에 융화되었다고 느끼게 했고 강의를 끝난 후에도 더 배우고 싶은 질문들을 제게 남겼습니다.

강의는 원어 수업으로 우리에게 조금은 어렵게 느껴질 수도 있는 영어로 진행되었습니다. 하지만 전반적으로 교수님께서 어려운 영어단어는 다시 쉬운 영어문장으로

로 풀어서 설명해주셨기 때문에 이해하는데 어려움이 전혀 없었습니다. 최대한 영어로 강의를 들으려고 노력했고 해석으로 인해 작품을 이해하는 데 큰 문제라고 생각될 때만 사전이나 번역기를 사용했습니다.

어떤 수업이든 저에게 작게나마 흔적을 남긴다고 생각합니다. 새로운 분야에 대해 궁금증을 남기거나, 해당 강의 내용을 심층적으로 공부하고 싶다는 열정을 남기거나. 어떤 모습으로든 선한 영향력이 되어 제가 좀 더 성숙한 모습으로 성장할 수 있도록 선순환을 반복시켜왔던 것 같습니다. ‘예술이란 무엇인가’ 라는 강의는 제게 큰 흔적을 남기게 될 것 같습니다. 수많은 강의와 과제 속에서 저는 ‘나라는 존재’에 대해 찾으려고 부단히 노력했고, 그런 뜻깊은 경험이 쉽게 잊혀질 것이라고 생각하지 않습니다. 비대면 강의로 모두에게 힘든 시기였지만 교수님의 철저한 수업 준비로 인해 많은 깨달음과 배움을 얻었습니다. 저에게 너무나도 소중한 강의였고, 대학 생활 중 잊지 못할 하나의 기억으로 남게 될 것 같습니다.

입선

비대면 수업에 대한 불만, <콘텐츠 기획의 인문학>으로 사라지다

교과목명 : 콘텐츠기획의인문학

교수명 : 홍경수

학과 : 문화콘텐츠학과

이름 : 김지영

1. 수업 참여 후기

문화콘텐츠학과는 하나의 콘텐츠를 제작하는 콘텐츠 기획자를 양성하는 학과이다. 그렇기에 이것에 있어 주를 이루는 ‘기획’은 실질적으로 이론을 배우는 것보다 직접 발로 뛰어 몸소 느끼는 실습 경험이 중요하다. 이런 학과에 입학하여 많은 경험을 할 준비가 되었었던 새내기였지만 코로나로 인해 작년 한 해 방구석에 앉아 이론을 배우는 것에 그쳤기에, 나는 상당히 많은 아쉬움을 느꼈다. 2021년에는 부분적으로나마 대면 수업이 진행되길 바랐지만, 2021년이 되어도 상황은 쉽게 완화되지 않았기에 2학년이 되었어도 비대면으로 수업을 듣게 되었다. 기대가 너무 컸던 걸까, 실망도 크게 느꼈던 나는 1년 넘게 비대면으로 ‘콘텐츠’를 배우는 것에 많은 불만을 가졌다. 이런 내게 있어 <콘텐츠 기획의 인문학>은 강의실에 발을 들여보지도 못하고 현내기가 되어버린 20학번의 불만을 잠재워주는 수업이었다.

문화콘텐츠가 인문학과 연관되어 있다는 것은 잘 알지만, 콘텐츠를 기획하는 데에 있어서 인문학이 어떻게 도움이 되는지는 알기 어렵다. 그러하여 콘텐츠 기획의 인문학 수업은 콘텐츠 기획에 도움이 될 만한 다양한 인문학적 사회학적 이론과 접근법을 검토하고, 이를 바탕으로 하나의 콘텐츠를 제작하는 수업이다. 수업시간에 배운 이론을 매주 개별 팀 토론 활동을 통해 적용시키고, 이런 연습의 과정을 거쳐 실제로 하나의 콘텐츠를 창작하는 것으로 수업이 마무리된다.

나는 이론을 학습함에 있어서 가장 중요한 것이 ‘기록’이라고 생각하는데, 이 수업에서는 매 수업마다 배운 이론에 대해 스스로 정리해보고 복습할 수 있는 기록물인 WIU (What I Understood)를 BlackBoard 토론 게시판에 올리게 된다. 이론을 기록한다는 것이 중요하다고 생각하였지만, 이론을 배우면서 어떤 생각이 들었는지,

어떤 점을 느꼈는지에 대해 적는 것은 생각보다 번거로워서 안 하게 되는 일이다. 하지만 WIU라는 과제를 수행하기 위해 이론을 정리함과 동시에 수업시간, 그리고 수업 후에 들은 나의 생각을 글로 풀어내어 하나의 ‘기록물’의 형태로 매주 남겨야 한다. 나의 생각을 글로 담아낼 때 확실히 나의 삶의 태도, 나의 목표가 드러날 수밖에 없었기에 공개적인 장소에 쓰는 글이지만 정말 솔직하고 진솔한 내 감정을 꼭꼭 담아내어 글을 썼었다. 그렇기에 15주간 매주 남긴 WIU에는 한 학기 동안의 나의 성장 과정이 고스란히 담겨있는 셈이며 동시에 교수님과 소통할 수 있는 창구가 되기도 하였다. 수업시간에 배운 내용을 자신이 어떤 식으로 이해했는지, 내가 어떤 마음가짐으로 교수님의 수업을 듣는지 등을 매주 교수님께 전달함으로써 비대면으로 이뤄진 교수님과 나의 사이가 좁혀지는 것을 느꼈다. 전공에 애착이 있는 나이기에 전공 교수님들과 많은 소통을 나누고 싶지만 매번 캡을 통해서만 짧게 소통을 해야 하는 것이 아쉬울 때가 많았다. 하지만 콘텐츠 기획의 인문학 수업에선 글로 한 번, 실시간 수업 때 또 한 번. 교수님과 소통할 수 있는 기회가 총 2번 주어졌기에 비대면이지만 교수님과 꽤나 가까운 사이가 된 것만 같아 좋았다. 교수님께서 실시간 수업 때 나의 WIU를 칭찬해주시면서 내 이름을 따뜻하게 불러주시는 순간에는 그렇게 기쁘지 않을 수가 없었다.

이론에서 기록이 중요하다면, 이를 적용시키는 실습에 있어서는 ‘협심’이 중요하다. 특히 콘텐츠를 만드는 것은 절대 혼자서는 할 수 없는 일이기에 여럿이 의견을 교환하고, 하나의 목표를 향해 모두가 힘을 합해 노력하는 것이 중요하며, 그렇기에 우리 과의 수업에선 팀플 과제가 많을 수밖에 없다. 콘텐츠 기획의 인문학에선 실시간 수업으로 이론에 대한 학우분들의 발표가 이뤄지고, 녹화 강의에선 조금 더 자세한 이론에 대한 정리, 심화 내용을 배우게 된다. 그리고 강의 말미에 심화 이론을 적용시킬 수 있는 팀 과제를 내준다. 이 팀 과제를 수행하기 위해 매주 주말마다 줌으로 조원들끼리 만나게 되는데, 3월에 구성된 조가 학기가 끝날 때까지 유지되기에 매주 팀원들을 만나게 되고, 실시간 강의 시간을 포함해 일주일에 2번은 꼬박꼬박 만나기에 더 친해질 수밖에 없었다. 더욱 친해지고, 의견 교환도 원활히 이뤄지고 과제물의 퀄리티 또한 점점 올라가는 게 느껴졌다. 이렇게 매주 회의를 하며 과제물을 만들어간 경험이 하나의 창작 콘텐츠를 기획하는 기말과제를 수행하는 데 있어서 큰 도움이 되었던 것 같다. 창의적인 의견을 내는 사람, 이러한 아이디어를 더욱 부풀려 확산시키는 사람, 이들의 의견을 잘 정리하여 아이디어가 사라지지 않게 하는 사람들이 매주 팀 과제를 수행하며 만들어간 협심이 최종 창작 콘텐츠를 제작하는 데 있어서 빛을 발한 것이다. 비대면으로 모든 회의를 진행하였기에 더 자주 만날 수 있었고, 사소한 문제도 시간을 금방 조율해 목소리를 공유하며 빨리 해결할 수 있었다. 이 덕분에 콘텐츠 기획에 있어서 중요한 능력인 ‘협

심'을 키울 수 있었다고 생각한다.

이렇게 콘텐츠 기획의 인문학 수업은 대면 수업에 대한 갈망이 가득한 20학번의 마음을 조금은 달래주는 수업이었다. 비대면 수업으로도 이렇게 즐거운 배움을 얻을 수 있고, 소통을 진행할 수 있다는 것을 깨닫게 되었다. 직접 얼굴을 보지 못한다는 것을 제외하고는 비대면 수업의 단점을 전혀 느낄 수 없는 수업이었다고 생각한다. 콘텐츠 기획의 인문학을 통해 나의 성장을 몸소 느낄 수 있었고, 교수님과 끈끈한 소통을 할 수 있었고, 학우분들과도 자주 소통하며 기획자의 자질을 키울 수 있었다.

2. 나만의 학습 노하우

14주차 WIU

“타이밍을 놓치지 마라”

“태양이 빛날 때 건초를 밀러라”는 서양 속담이 있듯이, 인생에서 가장 중요한 건 타이밍이라고 말한다. 인간은 여러 가지 변수가 존재하여 한치 앞도 알 수 없는 이 요지경 같은 인생을 살고 있다. 이런 예측불가 인생을 사는 인간에게 ‘타이밍’은 마치 귀한 선물과도 같다. 이러한 귀한 선물을 보다 잘 활용하기 위해선 자신에게 주어질 적절한 타이밍에 적절하게 대처할 능력을 미리미리 준비하는 것이 중요하다. ‘윈도 실패’라는 말이 있듯, 나는 ‘타이밍’도 실패라고 생각한다. (윈과 타이밍, 상당히 비슷한 말이긴 해도 뭔가 살짝 다른 느낌이다. 윈은 정말 예상치 못한 순간에 찾아오는 것. 타이밍은 뭔가 준비된 자에게 신이 내려주는 선물 같다는 생각이 든다.) 자신에게 주어질 타이밍을 얼마나 심분 활용하는지는 그 타이밍이 오기 전 인생에서 쌓아온 자신의 경험과 지혜에 달렸다. 타이밍은 주어진 자에게 기억할 일 찾아올 것을 나는 굳게 믿는다.

“하늘 아래 새로운 것은 없다”

여자들이 매우 공감할 말로 하늘 아래 같은 립스틱 색상은 존재하지 않는다”라는 말이 있다. 비슷한 색으로 보일지라도 정말 아주 미세하게 채도, 명도 등이 다르다. 그런데 이어서 피디님은 우리에게 하늘 아래 새로운 것은 없다고 말씀하셨다. 다소 당혹스러운 말이었다. 이는 콘텐츠 기획의 인문학에서 배운 콘텐츠 생산에 있어서 재개념과 관련된 말이었는데, 모든 유형의 콘텐츠는 재개념의 과정 속에서 알게 모르게 여러 유형의 콘텐츠 구성요소들이 혼집 되어있다. 그렇다면 나는 정말 신선하다고 생각되는 콘텐츠들도 사실 여러 콘텐츠의 구성요소를 교묘하게 잘 믹스하여 ‘새롭게 보이는’ 착을 하고 있는 것이 아닐까? 하는 생각이 들었다. 이 생각은 다소 위험한 생각이기도 하다. 나도 결국 ‘착해야 하는 사람이 될 것 같아서 말이다. 하지만 거짓말에도 선한 거짓말이 존재하는 것처럼 ‘착에도 선한 착이 존재하는 것이 아닐까, 하늘 아래 새로운 것은 없다는 상당히 혁신적인 말과 동시에 예비 콘텐츠 기획자에게 다소 험한 감정을 느끼게 하는 조언이었다.

콘텐츠 기획의 인문학 수업은 매주 무엇을 배웠는지, 수업에서 어떤 것을 느꼈는지에 대한 기록을 남기는 WIU라는 기록물을 남기게 된다. 그리고 교수님께서 일일이 다 읽어보시고, 우수하다고 생각되는 WIU를 4 ~ 5편 정도 수업 초반에 소개해주신다. 정확히 기억은 안 나지만 나의 WIU는 적어도 2주에 한 번씩은 우수 WIU로 채택되었기에 같이 수업을 듣는 선배들은 나를 향해 ‘교수님의 애제자’라는 별칭을 붙여주시까지 하였다. 수업시간에 배운 이론을 내 문장으로 정리하는 것뿐만 아니라, 이론을 배우고 이를 적용하는

용하는 팀 과제를 통해 어떤 점을 느꼈고, 어떤 생각을 하게 되었는지 상세히 기록하였기에 가능한 일이었다고 생각한다. 그리고 이것에 있어서 ‘녹화본 영상’을 여러 번 리플레이하는 것이 큰 도움이 되었다고 생각한다. 비대면 강의는 실시간 강의로 진행되는, 녹화 강의로 진행되는 ‘녹화본’이 존재하여 강의를 여러 번 돌려볼 수 있다는 큰 장점이 있다.

실시간 강의의 녹화본을 통해 이론을 천천히 귀 기울여 다시 들을 수 있을 뿐만 아니라 교수님께서 중간중간 학생들에게 질문을 하시고, 이에 대한 학우분들의 답변 또한 들을 수 있다. 본 수업시간에선 이론에 대해 집중하였다면, 녹화본 영상에선 학우분들의 의견에 조금 더 귀를 기울이게 되었다. 같은 이론을 배워도 정말 다른 사고로 접근하는 분들도 계시고, 자신의 의견을 정말 똑부러지게 얘기하시는 분

들도 계속해서 늘 수업을 통해 자극을 받게 된다. 이러한 자극은 나만의 신선하고 재밌는 WIU를 만들어내는 데 큰 도움이 되었다. 이러한 학습이 누군가에게는 귀찮음으로 다가오겠지만, 나는 비대면 학습의 장점을 최대한 잘 살릴 수 있는 학습 방식이라고 생각이 들었다. 한 학기 동안 진심을 다해 수업을 수강하였기에 정말 많은 것을 배우고 느낄 수 있었고 이 덕분에 좋은 학점 또한 받을 수 있었다고 생각한다.

입선

아는 만큼 보인다 (현실에서 이론이 적용되는 순간!)

교과목명 : 컴퓨터네트워크

교수명 : 노병희

학과 : 환경안전공학과

이름 : 박보현

1. 수업 참여 후기

[교수님과의 소통을 통한 원격수업의 한계 극복]

본 수업은 동영상 수업을 베이스로 하되, 교수님께서 학생들에게 의견을 물어 실시간 수업 주기를 정하고 2주에 한 번씩 실시간 수업을 하는 형식으로 한 학기 동안 진행되었습니다. 실제로 이 수업 방식이 매우 도움이 되었는데 평소에는 동영상으로 수업을 수강하고 실시간 강의 때는 동영상 수업을 들으면서 궁금했던 점을 질문하거나 교수님께서 다시 한번 내용을 정리하며 복습해주시는 방식으로 이루어졌습니다. 이런 수업 방식을 통해 원격수업의 한계인 소통의 부재를 충분히 극복할 수 있었고, 이 외에도 교수님께서 아주 BB의 Q&A 학습창을 통해 빠른 응답을 해주셨기 때문에 원격수업임에도 불구하고 학습하는 데 불편을 느끼지 못했습니다. 또한 중간에 약 2회 정도 Quiz를 봤고, 과제를 내주셨는데 이 부분에 있어서도 제출이 끝난 뒤 바로바로 정답과 점수를 올려주시고, 학생들에게 이의사항과 정답에 대한 문의를 받으시면서 답안을 인정해주시기도 하고, 틀린 부분에 대한 충분한 설명도 해주셔서 중간고사와 기말고사를 보기 전에 좀 더 확실하게 공부를 할 수 있었습니다.

[복합적인 개념 설명과 비교를 통한 학습]

이 수업은 컴퓨터네트워크의 OSI 7 Layers Model에 대해서 집중적으로 공부하는 과목이었는데, 7개의 Layer를 윗 Layer인 Application Layer부터 차례대로 설명하면서 진행되었습니다. Internet Protocol의 특성상 한 Layer가 독립적으로 작동하지 않고 위아래에 있는 Layer와 상호작용을 하며 작동하는 데 이런 부분에 있어서 교수

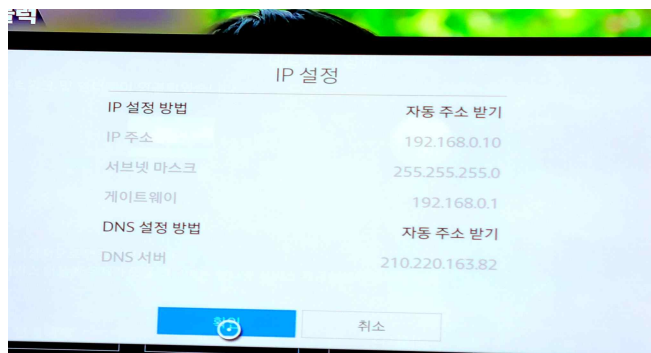
님께서 설명을 해주실 때, 다른 Layer와의 공통점과 차이점 비교, 또한 서로 어떻게 상호작용을 하는지에 대해서 꼼꼼하게 설명해주셔서 배웠던 내용들과 합쳐서 유기적으로 공부할 수 있었습니다. 또한 각 Layer을 공부하면서 다른 Layer들과의 관계가 궁금해지면, 스스로 검색을 통해서 찾아보기도 하고, 교수님께 질문을 해서 내가 잘못 알고 있는 오개념이 뭔지 정확히 알 수 있었습니다.

[장점을 최대한 활용할 수 있는 수업_ 반복 시청 가능한 수업 영상]

다들 원격수업의 단점만 부각하지만, 저는 원격수업의 장점도 있다고 말하고 싶습니다. 물론 대면 수업을 하는 거에 비해 소통의 부재가 생길 수 있고 집중도가 떨어질 수도 있지만, 그에 반해 대면 수업 때는 불가능했던 부분이 비대면 수업 시에는 가능해진 부분도 있습니다. 바로 수업을 이해할 때까지 여러 번 돌려 들을 수 있다는 점입니다. 또한 복습을 하기 위해서도 여러 번 들을 수 있고, 이해가 안 되면 잠시 정지를 시키면서 들을 수도 있습니다. 저는 특히 이번 컴퓨터네트워크 수업에서 이 장점을 많이 느꼈습니다. 이 수업의 경우 전공과목이고 정말 처음 접하는 내용들이 많아서 학기 초에 걱정이 많이 되었는데 이해가 될 때까지 반복적으로 학습을 하고, 시공간에 제약을 받지 않고 필요할 때는 언제든지 강의를 수강할 수 있는 시스템이었기 때문에 정말 여러 번 들으면서 한 학기 동안 학습했던 거 같습니다.

[아는 만큼 보인다!]

우리는 평소에 네트워크 속에서 살고 있기 때문에, 조금만 신경을 쓴다면 그 사실을 인지할 수 있습니다. 컴퓨터 네트워크 과목을 배우기 전에는 보이지 않았지만 이 과목을 배우고 나니, 정말 네트워크는 우리 실생활 곳곳에 있다는 것을 깨달았고, 예전에는 보이지 않던 네트워크 관련 사항들이 보이기 시작했습니다.

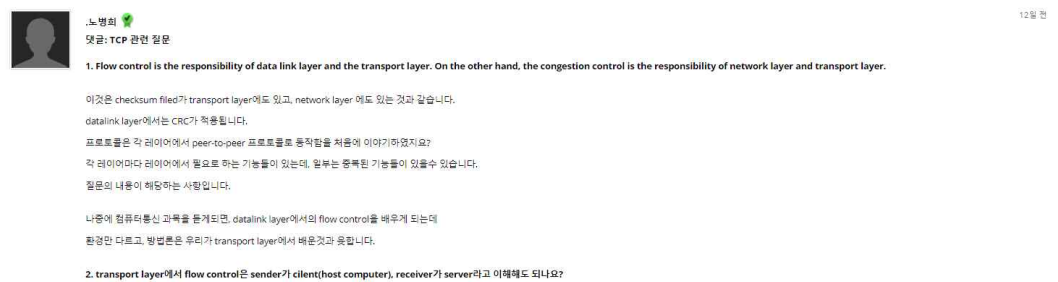
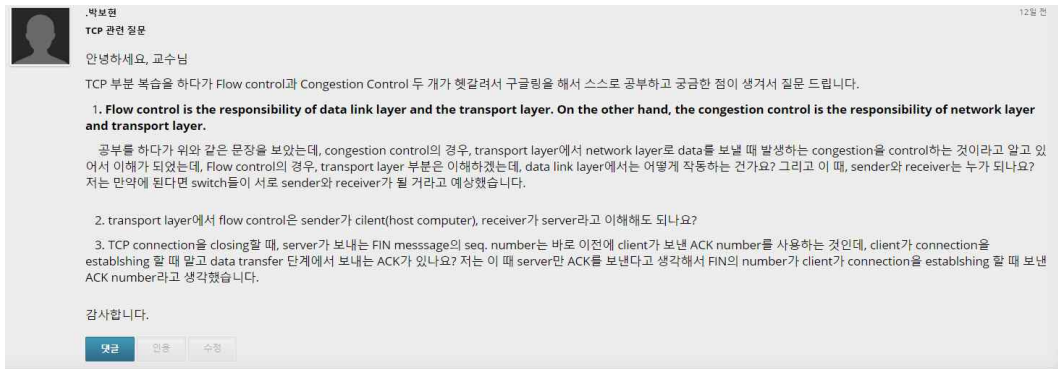


한번은 TV를 보다가, 스마트TV의 와이파이 주소를 받기 위해 연결을 하는데, 다음과 같이 이번에 배운 IP address 부분과 관련된 서브넷 마스크, DNS 서버, 게이트웨이 등의 CIDR 형태의 IP address가 나오는 것을 보고 다시 한번 배운 내용을 복기하며, 예전 같았으면 그냥 넘겼을 내용을 다시 한번 해석해보게 되었습니다. 이런 과정을 통해, 지루하게만 느껴지던 전공 수업이 실생활에서 적용 가능하다는 것을 알게 되었고, 전공 공부에 더 흥미를 느끼게 되었습니다. 정말 아는 만큼 보인다는 것을 다시 한번 깨닫게 된 계기였습니다.

2. 나만의 학습 노하우

[1] Q&A 적극 활용

수업을 듣다가 혹은 개인적인 학습 중 궁금한 점이 생기면, 아주 BB의 Q&A탭을 이용해 적극적으로 모르는 부분 혹은 헷갈렸던 부분을 질문하고, 교수님께서 답변을 달아주시면 본 내용을 토대로 다시 학습을 하는 식으로 공부를 진행했습니다. 이메일을 활용해도 되지만, Q&A를 통해 질문을 할 경우, 같은 수업을 듣는 다른 학우들도 제 질문을 볼 수 있고 답변을 달 수 있기 때문에 여러 가지 답변을 얻고 토론을 할 수 있어서 Q&A를 활용하였습니다. 또한 교수님께서 답변도 빨리 해주셔서 궁금한 점을 물어보고 금방 답변을 받아 학습하는 데 있어서 더욱 편하게 사용할 수 있었습니다. 아래에 Q&A를 어떻게 활용했는지 첨부합니다.

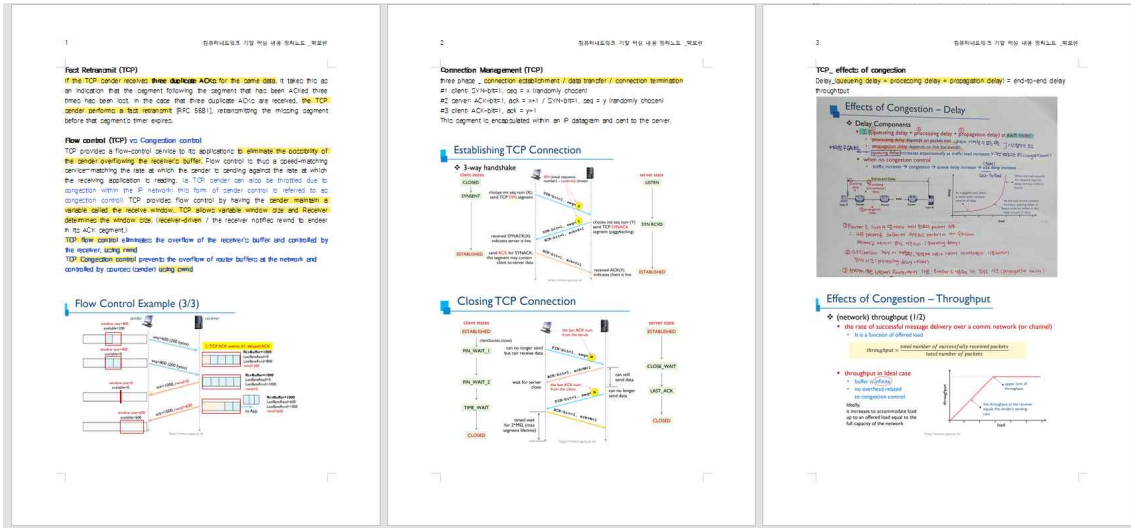


[2] 구글링을 통한 추가 학습

전공과목의 특성상, 그 깊이가 너무 깊어 종종 수업시간에는 간단하게 넘어가는 내용이 있습니다. 이런 경우, 교수님께서 이런 부분은 인터넷 혹은 논문을 찾아보고 학습한 뒤 질문을 하라고 하시곤 했습니다. 저는 이때, 구글링을 활용해서 인터넷에서 내용을 찾고 학습한 뒤, 강의노트에 필기를 하고 제가 학습한 게 맞는지 위에서 말한 Q&A를 통해 교수님께 질문드리곤 했습니다. 또한 이외에도 공부하다가 좀 더 많은 예시가 보고 싶거나, 궁금한 점이 생겼을 때 구글링을 통해 답을 찾곤 했는데, 이런 일련의 과정들이 쌓여서 학기 말이 되니 배운 개념을 한 번에 정립하는 데 큰 도움이 되었습니다.

[3] 나만의 학습정리 노트 만들기

시험 기간에는 주어진 강의자료도 활용하되, 저만의 정리 노트를 만들어서 공부하려고 노력했습니다. 첫 수업부터 마지막 수업까지의 강의내용들을 모두 하나씩 정리하며, 헷갈렸던 개념들은 좀 더 상세하게 내용을 적고, 비슷한 개념들은 두 개념 사이의 공통점과 차이점을 중심으로 정리를 했습니다. 이렇게 나만의 학습 정리 노트를 만듦으로써 다시 한번 모든 내용을 훑을 수 있는 좋은 기회가 되었습니다. 시험 하루 전에는 정리한 학습 노트를 토대로 공부를 하며, 한눈에 모든 내용들을 훑어볼 수 있었습니다. 다음과 같이 한 학기 내용을 약 25페이지로 요약했습니다.



[4] 주어진 학습 자료 및 기출 활용

교수님께서 공부할 때 이용하라고 다음과 같이 강의자료와 Quiz, 중간, 기말 기출문제를 올려주십니다. 올려주신 기출문제의 양이 생각보다 많기 때문에 다 풀지 못할 수도 있지만, 이런 부분에 있어서 특히 저는 무조건 다 풀고 틀린 부분은 확실하게 오답을 하며, 제가 공부했던 내용들을 확실하게 알고 있는지 꼭 확인하려고 했습니다. 교수님께서 올려주시는 기출 자료만큼 공부하고, 공부한 내용을 확인하는 데 훌륭한 자료는 없다고 생각했습니다. 따라서 년도별, 개념별로 문제를 정리하고 풀어서 실제로 도움을 많이 받았습니다.

강의노트

- 강의자료
- 강의녹화2021
- 중간고사 기출문제
- 기말고사 기출문제
- Quiz 1 기출 문제
- Quiz 2 기출문제

입선

어찌할 수 없는 영역인 코로나 VS 어찌할 수 있는 영역의 ‘나’

교과목명 : 수학2

교수명 : 박성웅

학과 : 산업공학과

이름 : 박서혜

1. 수업 참여 후기

작년부터 시작된 팬데믹 사태로 모든 교육기관이 원격수업을 진행해왔습니다. 이에 모두에게 같은 상황이 주어졌을 때, 외부의 상황은 절대 나의 결과에 대한 원인과 핑계가 될 수 없다고 생각했습니다. 또한 평소 ‘세상에는 내가 통제할 수 없는 게 투성이인데, 내가 통제할 수 있는 영역마저 통제하지 못하면 뭘 할 수 있겠느냐’ 라는 저만의 신조를 가지고 있었기 때문에 저의 학습에 외부 요소가 방해물이 되지 않도록 저를 통제하기 위해 노력했습니다. 일단 저는 수강 신청에 있어서 선택이 가능한 과목들은 녹화 강의 수업이 아닌 실시간 ZOOM 수업을 하는 수업을 위주로 수강 신청을 했고, 부지런하고 규칙적인 생활을 위해 월 ~ 금 모두 9시 수업으로 수강 신청을 하였습니다.

교수님 중에서는 수업을 하실 때, 수업 내용 전달을 우선시 여기시는 교수님들도 있으시고, 소통을 중시하여 ZOOM 혹은 구글 MEET의 채팅창에 대답을 요하시거나 마이크를 켜고 소통하기 혹은 학생들의 화면까지 켜고 수업하기를 원하시는 교수님들도 계십니다. 제가 들은 수업 중 박성웅 교수님의 ‘수학2’ 수업에서는 수업이 시작되면 가장 먼저 전자출결 번호를 불러 출석 확인을 하신 후에 수업 참여가 가능한 학생들에게 ZOOM의 ‘손들기’ 버튼을 누르라고 하셨습니다. 수업 참여는 교수님께서 수업시간에 한 명씩 호명하시면 학생이 마이크를 켜고, 교수님과 1대 1로 과외하는 듯한 방식으로 진행되었습니다. 저는 이 방식이 제가 언제 호명될지 몰라 긴장도 되었고, 제 차례가 되었을 때는 수업 중 이해가 안 되는 부분을 바로 질문할 수 있어서 좋았습니다. 또한 제 이름이 수업 막바지쯤 호명되면 수업 후, 연습문제 혹은 예제를 풀다가 막힌 문제들에 대해 수업 후 설명해 달라고 부탁하기도 하였습니다. 기초 수학 과목들은 아주 BB의 조직의 Q&A란에 질문을 할 수 있지만, 그래도 음성과 화면으로 설명을 바로 들을 수 있어서 공부하는데 큰 도움이 되었습니다.

니다.

교수님의 열면 수업과 저의 적극적인 수업 참여 노력으로 비대면 수업임에도 학습 환경이 저의 학업에 영향을 준다는 느낌을 받지 못하였고, 만족스러운 결과도 뒤따랐습니다.

2. 나만의 학습 노하우

저의 수학2 수업을 중간고사 전과 후로 나누어서 설명을 해보겠습니다. 저에게 대학에서의 비대면 수업은 올해가 처음이었습니다. 하지만 ‘1번 문항의 수업 참여 후기’에 쓴 것과 같이 제가 통제할 수 있는 영역은 되도록 통제하기 위해 수업시간에 손들기도 열심히 하고, 졸리면 공부하던 자리에 서서 수업을 듣기도 하였습니다. 하지만 중간고사 전에 본 반별 1차 퀴즈에서 30점 만점에 19점이라는 다소 아쉬운 점수를 받았습니다. 점수라는 피드백을 받은 후, 중간고사 이후에는 수업 듣는 환경을 더 통제하고, 공부 방법에도 변화를 주어야겠다고 생각했습니다.

먼저, 공간을 통제하기 위해서 수업 듣는 장소를 고정하는 방법을 사용하였습니다. 아무래도 집이라는 감시가 없는 비교적 자유로운 공간에서 수업을 들었었기 때문에 식탁에서 식사 후, 후식을 먹으면서 노트북이나 태블릿을 켜서 수업을 듣기도 하고, 혹은 거실에서 아니면 내 방 책상에서 수업을 듣는 등 그때그때 편한 방식으로 수업을 들었습니다. 하지만 중간고사 이후에는 수업을 제 방 독서실 책상에서 ‘만’ 듣기로 규칙을 정하니 다른 걸 하다가도 수업시간에 방문을 열고, 지정된 책상에 앉는 행위 자체가 마치 지정된 강의실에 앉아서 수업을 듣는 것과 같은 의식처럼 여겨졌습니다.

그리고 저는 학습 방법에서도 변화를 주었습니다. 물론 시험에도 문제가 영어로 나오기 때문에 원서에 나온 단어로 공부해야 하고 내용을 이해해야 하는 것은 맞지만 용어의 한글 혹은 한자 뜻도 알면 이해가 더 잘되는 개념들도 있었고, 용어를 매번 까먹었기에 용어만 따로 정리하여 문제에서 어느 개념을 묻는 것인지 바로 바로 파악이 가능하도록 정리하였습니다.

또한 기말고사 시험 범위에 해당하는 스칼라 적분, 벡터 적분, 매개변수를 활용하여 경로적분, 밀도 구하기 등 개념 각각을 배울 땐 어렵지 않았는데, 막상 웹과제를 풀다 보니 어떤 문제에 어느 개념을 적용해야 하는지조차도 파악이 안 되어서 문제를 접근하는 것조차 어렵다고 느낀다는 것을 파악했습니다. 다행히도 웹 과제에는 해당 개념이 어디 파트에서 설명된 개념인지 문항 옆에 단원 번호가 적혀 있었습니다. 따라서 어떤 부분을 보고 어떤 개념을 적용해야 하는지부터를 공부하였습니다.

마지막으로, 기출문제를 풀면서 문제에 잘 접근해도 계산 실수를 해서 틀리는 문

제가 생각보다 많다는 것을 발견했습니다. 그래서 자주 틀리는 적분 계산 부분의 문제들은 조직의 강의계획서에 나온 추천 예제에 더하여 주변 문제들까지 추가로 풀어보면서 계산 실수를 줄이도록 하였습니다.

이러한 노력으로 기말고사 전에 보는 반별 2차 퀴즈에서는 반에서 혼자만 만점인 30점을 받고, 기말고사에서는 150명 이내의 학생들 중 17등을 하는 쾌거를 이룰 수 있었던 것 같습니다.

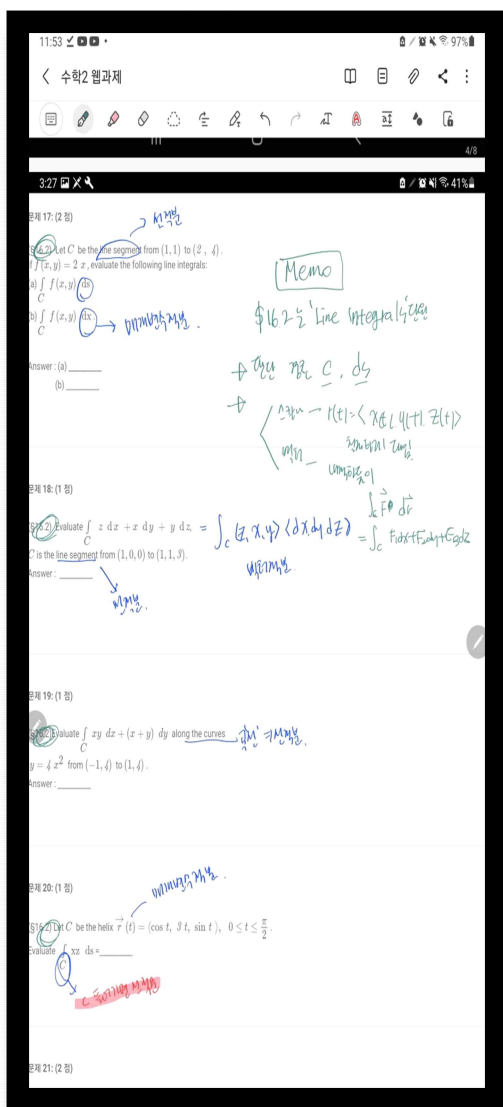


사진1

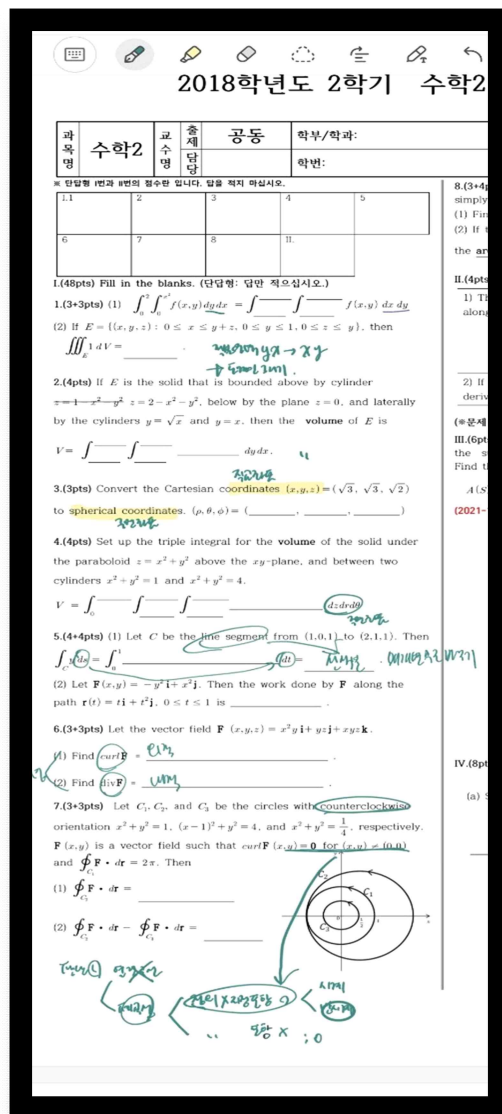


사진2

사진1은 웹과제에서 어떤 부분을 보고, 어떤 개념을 사용해야 하는지는 파란색 펜으로, 단원의 주요 개념은 초록색 펜으로 정리하여 문제 접근법을 익히고 시험 전에 복습하였습니다.

입선

비대면으로도 가능한 양방향적 소통

교과목명 : 내분비

교수명 : 김혜진 외

학과 : 의학과

이름 : 박소연

1. 수업 참여 후기

코로나19가 본격적으로 전파되기 시작한 2020년 초는 전 세계적으로 많은 혼란을 겪은 시기입니다. 이런 혼란스러운 상황에 적응할 새도 없이 코로나19의 여파로 갑작스럽게 수업이 비대면으로 전환되었습니다. 학교는 등교 중지되고, 수업은 전부 녹화 강의로 진행되었습니다. 수업이 녹화되어 아주 bb에 업로드되었기 때문에 같은 수업을 여러 번 다시 들을 수 있는 장점을 누릴 수 있었습니다. 그러나 커뮤니케이션 측면에서는, 소통이 너무 일방향적으로 이루어졌습니다. 녹화 강의를 통해 일방향적으로 지식이 전달되었고, 학생들은 그저 그것을 받아들일 뿐이었습니다. 대면 수업과 달리 궁금한 점이나 이해가 되지 않는 부분이 생겼을 때 바로바로 질문할 수 없었기 때문에 불완전한 지식을 쌓아 올리는 느낌이었습니다. 또한 발표, 토론, 실습과 같은 과정을 통해 배운 지식을 인출하고 응용할 수 있는 기회가 거의 없었기 때문에 배운 지식은 얼마 지나지 않아 머리에서 휘발되었습니다.

그러나 올해는 작년과 달리, 양방향적 소통이 가능해졌습니다. 특히 내분비학 과목의 경우 비대면 수업임에도 불구하고 여러 장치들을 통해 대면 수업 못지않게 수준 높은 학습 효과를 누릴 수 있었습니다. 우선, 내분비학 과목에서 처음으로 실시간 줌 Q&A 세션이 생겼습니다. 학생들이 녹화 강의를 들은 후, 녹화 강의를 통해 해결하지 못한 궁금증들을 실시간 줌 Q&A 세션을 통해 질문할 수 있었습니다. 이에 따라 일방향으로 국한되어있던 소통이 양방향으로 확장될 수 있었습니다. 또한, 증례 발표 과제가 주어져 학생들이 배운 내용을 응용할 수 있는 기회가 생겼습니다. 이 과제를 통해 학생들이 각자 하나의 증례를 맡아 환자의 증상, 활력 징후를 분석하여 진단을 내리고, 그에 따른 치료 방법을 제시하는 발표를 했습니다. 배운 지식을 그저 암기하는 것이 아니라 그것을 응용하여 유의미한 내용을 창출해내는 과정을 거침으로써, 배운 내용을 탄탄히 다지고 본인의 것으로 만들 수 있었습니다.

또한 내분비학 과목의 수업 과정에는 교수님들로부터 내분비학 지식을 배우는 수업뿐만 아니라 외부 연사님들의 초청 강의 시간도 포함되어 있었는데, 이 시간에도 연사님들과 소통하며 강의로부터 많은 것을 배울 수 있었습니다. 김미영 연사님께서 제1형 당뇨병 환자의 가족으로 사는 것에 대한 이야기를 해주셨고, 강성지 연사님께서 디지털 헬스케어와 연관된 내분비학에 대한 강연을 해주셨습니다. 초청 강의를 실시간 줌으로 진행되었는데, 연사님께서 강연 중간중간에 퀴즈를 내시면서 학생들의 적극적인 참여를 유도하셨습니다. 또한 강연 마지막에는 연사님께 질문할 수 있는 시간이 마련되어 있었습니다. 이로 인해 초청 강의 시간 동안 연사님과 학생 간의 적극적인 소통이 이루어질 수 있었고, 적극적인 소통이 이루어질 수 있었기에 강연으로부터 많은 것을 배우고 느낄 수 있었습니다. 단순히 내분비학에 대한 의학적 지식만을 배우는 것을 넘어, 의학의 경계를 넘어 다른 분야에서는 내분비학이 어떻게 적용되고 어떤 역할을 하는지 깨우칠 수 있었습니다. 내분비학이라는 학문을 다양한 관점에서, 다차원적으로 이해할 수 있게 해준 의미 있는 시간이었습니다.

2. 나만의 학습 노하우

1) 실시간 Q&A 세션에 활발히 참여하기

교수님께 실시간으로 질문을 할 수 있는 실시간 Q&A 세션 때 적극적으로 질문하고 다른 친구들의 질문도 경청함으로써 학습 효과를 높일 수 있었습니다. 가령 저의 경우, 노태훈 교수님의 '시상하부-뇌하수체 질환의 신경외과적 치료' 녹화 강의를 들은 후 hook effect에 대한 궁금증이 생겼습니다. Adenoma가 있는 환자의 경우 prolactin이 정상 수치인 경우가 있는데, 이것은 prolactin이 너무 높으면 오히려 수치가 정상이거나 낮게 나오는 hook effect 때문이고, 그렇기 때문에 의심을 하는 것이 중요하다고 교수님께서 녹화 강의에서 말씀하셨습니다. 저는 이 부분을 듣고 그렇다면 어떤 경우에 의심을 할 수 있는 것인지에 대한 궁금증이 생겼습니다. 따라서 녹화 강의를 들은 직후 진행된 실시간 Q&A 세션에서 이 부분에 대한 질문을 하였고, 교수님께서 바로 답변을 해주셨습니다. 이처럼 질문이 머릿속에서 사라지기 전에 바로 실시간으로 질문함으로써 해당 수업 내용의 지식을 탄탄히 쌓을 수 있었습니다. 또한 다른 학생들의 질문도 경청하면서 제가 놓쳤던 부분을 확실히 짚고 넘어갈 수 있었습니다.

2) 발표 기회를 적극적으로 활용하기

내분비학 과목에서는 환자 증례에 대해 발표하는 과제가 주어졌는데, 이 기회를 적극적으로 활용함으로써 학습 효과를 높일 수 있었습니다. 원래 수업 과정의 대부분은 녹화 강의로 구성되어 있어 학습 방식이 교수님께서 수업하신 내용을 그대로 받아들이는 식이었습니다. 따라서 지식을 인출하는 과정이 없었고, 그저 지식을 습득하는 과정만이 주로 이루어졌습니다. 그러나 발표 과제를 통해 배운 지식을 직접 인출하고, 응용해보고, 고민해볼 수 있는 기회가 생겼습니다. 이 기회를 놓치지 않고 적극적으로 활용함으로써 학습 내용을 복습하고, 더 나아가 배운 지식을 인출하여 지식을 탄탄히 다지는 시간을 가질 수 있었습니다. 우선 제가 맡은 환자 증례를 꼼꼼히 분석했습니다. 환자의 증상, vital sign, lab test 결과 등을 보면서 impression을 잡았습니다. 증례 분석 후 환자 증례와 관련 있다고 생각되는 수업들을 모두 복습했습니다. 복습을 하니 처음에는 뚜렷하게 보이지 않던 환자의 진단이 명확해졌습니다. 배운 내용을 통합적으로 생각하고 고민하여 우선 대사 증후군으로 대략적인 진단을 내릴 수 있었습니다. 이후 진단을 확실히 하기 위해 환자의 검사치 중 대사 증후군의 기준에 합당한 요소들을 찾았습니다. 증례에 제시된 정보 중 대사 증후군의 진단 기준에 해당하는 허리둘레, triglyceride 농도, hdl 농도, 혈압, 고복혈당만 남기고 나머지는 제외시켰습니다. 이 다섯 가지 기준들을 분석하며 대사 증후군의 진단 기준을 충족하는지 살펴본 결과, 다섯 가지 기준이 모두 만족되어 대사 증후군으로 확진할 수 있었습니다. 진단 후, 더 나아가 대사 증후군의 병인과 치료방법도 발표에 제시하였습니다. 이렇게 발표 과제를 마치고 난 후, 이전에는 대략적으로만 이해했던 대사 증후군에 대해 훨씬 더 전문적으로 이해할 수 있었습니다.

3) 초청 강의 시간에 연사님과 소통하기

초청 강의 시간에 그저 수동적으로 강연을 듣는 것이 아니라 연사님과 적극적으로 상호작용함으로써, 강연에 더욱 집중할 수 있었고 강연으로부터 배워가는 것도 더 많아졌습니다. 제1형 당뇨병 환자 가족으로 사는 것에 대해 강연해주신 김미영 연사님께서 강연 중간중간에 퀴즈를 내셨는데, 이에 능동적으로 참여하고 연사님과 소통함으로써 당뇨병 및 그 환자들의 삶에 대해 더욱 깊게 배울 수 있었습니다. 또한 '디지털 헬스케어와 내분비학'을 주제로 강연하신 강성지 연사님의 강연을 듣고 나서는 궁금한 점이 생겨 질문 시간에 질문을 했습니다. 동기들과 다른 길을 가는 것에 대한 두려움은 없으셨는지, 있다면 어떻게 극복하셨는지에 대해 질문하였는데, 이에 대해 연사님은 남들이 가는 길을 가는 것이 아니라 자신이 가고 싶은 길을 가는 것이 진정한 나 자신의 인생을 살 수 있는 길이라고 말씀해주셨습니다. 이처럼 초청 강의 시간에 연사님과 소통함으로써 학문적 지식과 더불어 인생 조언까지 얻을 수 있었습니다.

입선

비전공자의 교양 도전기

교과목명 : 국제관계의이해

교 수 명 : 김홍규

학 과 : 미디어학과

이 름 : 이지수

1. 수업 참여 후기

정말 아무것도 모르는 상태로 수업에 참여했다. 뉴스에서 흘러나오는 정치 이야기는 어른들의 영역이라고 생각하며 모르는 척했다. 하지만 이제는 그런 변명이 통하지 않는 나이가 되었다. 소중한 한 표를 행사할 수 있는 어른이 되었기 때문에, 내가 살고 있는 이 나라와 이 나라를 둘러싼 나라들이 어떻게 돌아가는지 알아야 한다는 생각이 들었다. 혼자서 뉴스를 보면서 분석하기에는 모르는 것이 너무 많아서 신청한 수업이었다.

같이 수강하는 학생들이 10명이 채 되지 않아서 폐강되는건 아닌지 걱정이 많았다. 하지만 다행히(?) 폐강이 되지 않았다. 학생들의 수가 적었기 때문에 각자의 의견을 모두 말할 수 있었고 토론처럼 질문이 오고 갈 수 있었다.

국제관계와 관련된 책을 한 권씩 맡아 요약한 것을 발표하고 핵심 질문 하나를 선정하고 대답하는 식으로 수업이 진행되었다. 첫 주 동안에는 교수님께서 간단하게 국제관계와 관련된 용어를 설명해주셨다. 거의 모든 것들을 처음 들어봐서 필기하기 바빴다. 교수님께서서는 단순 용어 설명이 아니라 어떻게 국제관계 관련 책을 읽고 사고해야 하는지, 자신만의 의견을 어떻게 설명해야 하는지 구체적으로 설명해주셨다. 특히 한 주제를 가지고 각자의 의견을 물어보셨는데 그때마다 논리의 허점이나 발표를 할 때 고쳐야 할 점을 한 사람 한 사람 알려주셨다. ZOOM으로 수업이 진행되었는데 발표를 하거나 말을 할 때 올바른 자세에 대해 알려주셨고, 나는 미처 알지 못했던 발표 시 안 좋은 버릇을 확인해서 알려주셨다.

일주일에 한 번은 학생들의 발표와 토론이었고 또 다른 한 번은 교수님께서 정해 주신 주제와 관련된 영상자료나 자료(논문, 기사 등)를 보면서 서로 의견을 나누는 방식의 수업이었다. 현재 정세와 관련된 내용이라서 뉴스를 보면서 궁금했던 문제에 대해 깊이 생각해볼 수 있는 시간이었다.

시험은 다른 과목처럼 정답이 정해져 있는 문제가 아니라 자신의 의견을 논리적으로 적는 것이었다. 미리 문제를 공지해주셨고 그에 대한 여러 자료를 읽어보고 자신의 의견과 적절한 답안을 적었다. 중간고사, 기말고사 문제는 현재 정세와 관련된 문제들이었다. 늘 아빠만 보시던 뉴스를 어느새 옆에서 같이 보게 되었고, 오락 거리로만 이용하던 유튜브를 자료를 얻기 위한 수단으로 사용하기 시작했다. 중간, 기말에서 언급된 문제들의 과거까지 알고 싶은 마음에 도서관에서 관련된 책을 빌려서 읽었다. 교양이었음에도 계속해서 궁금한 점이 생겼고 그에 대한 답이 학자마다, 사람마다 달랐기 때문에 계속 책을 읽었다. 덕분에 나는 국제관계에 대해 관심을 갖고 뉴스를 살펴보고 현재 정세를 조금이나마 파악할 수 있는 사람이 되었다.

2. 나만의 학습 노하우

국제관계나 정치에 대해 아무것도 알지 못한 상태라서 첫 번째 수업부터 막막했었다. 교수님께서 상세하게 설명해주셨음에도 계속 궁금한 점이 생겼다. 특히 국제관계 같은 경우 매일 변화하고 뉴스를 통해 새로운 사건이 흘러나오기 때문에 현실을 따라가려면 따로 공부를 해야 할 것 같다는 생각이 들었다. 발표 순서가 정해져 있고 책 목록을 첫 수업 때부터 보여주셨기 때문에 수업 전 주부터 관련된 책을 도서관에서 빌렸다. 우선 책을 한 번 꼼꼼히 읽었다. 그리고 목록을 보면서 이해가 가지 않았던 부분들을 다시 읽었다. 책은 출판하는 기간이 있기 때문에 아무리 최신책이라고 해도 현재 정세와 다를 수밖에 없었다. 그래서 현재 정세와는 어떻게 다른지 검색을 통해 알아보았다. 그리고 수업을 하기 전날 발표자가 올려놓은 자료를 살펴보았다. 여태 읽었던 책들과 발표하게 될 책 사이에 어떤 차이점이 있는지 확인하고, 그것들이 현재와 어떻게 다른지 확인했다. 발표자가 제시한 질문에 나는 어떻게 대답할 수 있는지 미리 생각해보고 메모했다.

중간고사와 기말고사는 교수님께서 내주신 문제에 대한 답과 자신의 의견을 적는 것이었다. 미리 공지를 해주셨기 때문에 자료를 충분히 찾아볼 수 있었다. 우선 중간고사 문제와 관련된 정보를 모으기 시작했다. 논문, 기사, 책 등 관련된 정보가 있는 자료는 모두 기록했다. 항상 도서관에서 정치 분야 관련 책들은 쳐다보지도 않았는데 이번 학기는 유독 정치 분야 책들을 많이 살펴보았다. 우리나라 정치, 안보와 관련하여 많은 자료들을 읽고 내 생각을 정리했다. 새롭게 알게 된 내용도 많았고 문제점을 파악할 수 있었다. 이러한 문제점을 어떻게 해결할 수 있는지 생각하게 되면서 중간고사와 기말고사에 대한 나만의 답을 찾아 나갔다.

내가 발표 순서가 되었을 때는 발표 2주 전부터 여유롭게 준비했다. 중간고사 이후 발표였기 때문에 더 여유롭게 준비할 수 있었지만 다른 강의와 과제가 많았기

때문에 미리 준비했다. 일단 발표할 책을 읽었다. 책을 읽은 다음, 챕터 별로 내용을 요약했다. 발표할 책이 2014년에 출간된 책이었기 때문에 현재와 다른 것들이 많았다. 중국에 대한 전반적인 내용이 담겨있는 책이었는데, 현재는 달라진 제도부터 국가 간의 관계까지 달라진 점들을 정리했다. 책을 요약한 것만으로도 충분히 많은 내용을 담았기 때문에 따로 보충자료를 만들어 현재와 책이 어떻게 다른지 정리해서 발표를 듣는 사람의 이해를 높였다.

국제관계의 가장 큰 특징은 매일 변하는 것이라고 생각한다. 이 학문을 공부하는 입장에서는 이 특징이 큰 장점일 수도, 단점일 수도 있다. 시시각각 변화하는 정세에 무엇을 공부해야 할지 막막하지만, 바로 이론을 적용해보고 생각할 수 있다는 장점이 있다. 나 역시 이 강의를 듣고 난 뒤로부터 다양한 책을 읽게 되면서 나만의 의견이 만들어졌다. 뉴스를 보면서 흘러나오는 이야기를 듣고, 앞으로 정세에 대해 예측하고 그 이면에는 무엇이 있는지 생각해보게 되었다. 배운 것을 바로 적용하고 예측함으로써 국제관계에 대한 학습을 완성시켰다. 그저 앵커가 말하는, 아나운서가 말하는 내용들을 듣고만 마는 것이 아니라, 내가 그동안 정리한 내용들을 바탕으로 앞으로의 정세를 예측해보는 것이었다. 실제로 기말고사 문제는 내가 기말고사 전부터 생각해오던 주제와 관련된 것이었다. 이번 학기에 계속해서 책을 읽고 나만의 핵심 질문을 찾아내고 대답하는 과정을 통해 국제관계를 더욱 깊이 이해할 수 있었다. 발표 내용에 대한 질문을 서로 주고받으면서 새로운 의견에 대해 알게 되었다. 아무것도 모르는 상태로 시작했던 터라 책을 읽게 되면서 알게 되는 새로운 내용들과 교수님과 함께 하는 수업이 내가 주도적인 학습을 할 수 있도록 해 준 것 같다.

입선

부족한 환경 안에서 최고의 학습 시스템 구현

교과목명 : 정보시스템기초

교수명 : 박기진

학과 : 산업공학과

이름 : 조성호

1. 수업 참여 후기

COVID-19 사태에 따라 각 단체에서 그에 대한 대안을 내놓은 상태였다. 내가 다니는 학교인 아주대학교 또한 그러한 대안으로 비대면 수업을 택한 상태였다. 그렇게 첫 비대면 수업을 시작하였는데, 나는 화가 났다. 이는 단순 인터넷 강의와 다를 것이 없지 않나 싶었고 나의 등록금이 아깝게 느껴질 정도였다. 학교에서 제공하는 시설을 이용할 수 없게 되었고, 녹화 강의를 듣게 되어 교수님과의 커뮤니케이션이 아예 없다고 느껴져 이는 배우는 ‘학습’ 이라기 보다는 그냥 듣고 넘어가는 ‘이수’ 로 밖에 느껴지지 않았다. 하지만 그러한 수업들 중 박기진 교수님께서 가르치시는 ‘정보시스템 기초’ 라는 과목은 매우 달랐다.

보통 3학점을 받는 수업이라면 일주일에 3시간의 강의를 있는 것이 맞지만, 이 수업은 3시간의 녹화 동영상 강의와 1시간 30분의 실시간 강의로 이루어져 있다. 그런데 특별하게 느꼈던 것이 있다. 실시간 강의를 할 때 다루는 내용이 새로 배우는 것이 아닌 교수님께서 올려놓은 녹화 강의에 대한 복습을 해주는 방식과 학생끼리의 토론을 활성화를 시키는 수업이다. 이 방법은 학생에게 있어 매우 큰 영향을 준다.

첫째, 녹화 강의를 듣지 않은 학생은 실시간 강의에 ‘참석’ 을 한다 해도 ‘참여’ 를 할 수 없다.

둘째, 녹화 강의를 완전히 이해하지 않고, 단순히 시청만 한다면 이 또한 참여를 할 수 없다.

셋째, 녹화 강의에서 던지시 이야기한 사례를 직접 탐방하며 실제 사례에 대해 알 수 있는 토론 수업이 있어, 개념에 대한 이해뿐만이 아니라 실제로 적용하는 내용을 다뤄 실제 필요한 상황이 생겼을 때 학생이 다룰 수 있도록 도와준다.

이러한 수업 방식은 정말 획기적이라는 생각이 들었다. 학생이 공부를 할 수밖에

없는 상황을 만들어, 수업을 이수하는 거의 모든 학생들의 개념적 지식과 능력을 끌어올릴 수 있는 방법이라고 생각을 하였다.

녹화 강의를 들으며 개념을 이해하려 하고, 실시간 수업에서의 복습을 통해 나의 개념을 확인하고, 실사례 토론을 하여 실제 상황에 사용할 수 있게 하는 3단계로 배우는 완벽하다고 느끼는 교육방식이였다. 정말로 수업을 들으면서 잘 배웠다고 말할 수 있는 수업이였다. 하지만, 이 수업은 이러한 장점만 있는 것이 아니였다.

‘SLACK’이라는 교수님과 학생 간의 소통할 수 있는 애플리케이션을 이수하는 모든 학생들이 다운을 받아야 하는데, 이를 통해 다른 학생들이 올린 질문을 답변을 할 수 있는 학생들이 답변을 하거나 답변자가 없으면 교수님께서 답변을 해주는 방식으로 ‘SLACK’을 활용하였다. 나는 이것이 매우 큰 장점이라 생각했는데, 대면 수업을 할 때 보통 모르는 문제가 있으면 교수님이나 친한 친구에게 질문을 하여 답변을 찾아낼 수 있다. 하지만 교수님께서 이러한 시스템을 구축하여, 나와 친하지 않아도 답변을 할 수 있고, 다른 사람이 궁금해한 질문을 알 수 있어, 생각도 안한 부분에서 깨달음을 느낄 수 있어 또 한 번의 배움의 기회가 생긴 것이다.

이처럼 21년도 1학기에 배운 ‘정보시스템기초’라는 과목은 교수님의 노력과 다양한 시스템의 구축으로 대면 수업의 큰 장점인 학생과 교수 간의 활성화 된 소통 기능 못지않은 소통 능력을 발휘하였고, 교수님의 실시간 강의 수업 방식으로 학생들은 어떤 일이 있어도 녹화 강의를 듣고 나서 참여를 하게 되어 학생들의 참여율을 매우 높였다. 정말로 비대면 상황이 온 위기의 시기인데도 양질의 수업을 들을 수 있어 ‘나는 축복받은 사람이구나.’라는 느낌을 받을 수 있었던 수업이였다.

2. 나만의 학습 노하우

일단 녹화 강의를 들을 때, 나는 PPT에 있는 모든 내용을 필기를 하였다. 그리고 필기를 하며 직관적으로 이해가 되지 않는 부분에 체크를 하여 인터넷 검색을 통해 이해가 될 때까지 찾아보고 생각을 하였다. 이런 방법이 효율적이지 못하다고 생각이 들 수 있겠지만 찾아보며 허탕을 치게 된 정보들까지 머리에 들어오게 되며 다른 부분들도 배워 전체를 보는 시야를 가질 수 있게 된다고 생각한다.

그러한 노력을 했음에도 불구하고 정말 모르겠다고 느끼는 것은 교수님께서 이용하라 하신 ‘SLACK’에 질문을 올려 다른 학생들은 어떻게 생각하는지 물어보고, 교수님과 학생들에게 답변을 받아내어 다시 생각을 정리해 이해를 하였다.

그리고 교수님께서 수업에서 언급한 사례 등을 직접 찾아보기도 하였다. 이러한 사례들이 차후에는 토론 수업에 사용될 것을 알아 미리 예습을 한다면 나중에 실시간 수업 시 더 높은 효율을 끌어낼 수 있다고 판단이 되었다.

또한 내가 질문이 없어도, 일주일에 한번은 ‘SLACK’에 들어가 다른 학생들은 어떤 질문을 하였는지 확인을 하고 내가 아는 선에서 답변을 하였다. 다른 학생들의 질문을 찾아본다면 내가 미처 확인하지 못한 부분도 알 수 있게 되어 더 높은 학습 효율을 이끌어낼 수 있다.

마지막으로 토론 수업 시 발표를 도맡아서 하려고 노력을 하였다. 보통 학생들은 발표하는 것을 꺼려하는데, 물론 나는 꺼려하지 않는다. 발표를 직접 준비를 하며 그 과정에서 생각 정리도 할 수 있고, 궁극적으로 토론한 내용을 나의 머릿속으로 가져올 수 있기 때문이다.

이러한 공부 방식으로 수업을 들었다. 물론 학점을 잘 받을지 못 받을지는 나의 성적이 어떻게 나왔느냐에 따라 다르겠지만 학점과는 별개로 정말 많이 배운 수업이라 만족도가 제일 높았던 것 같다.

입선

주어진 환경을 받아들이고, 극복하기까지

교과목명 : 일본어1

교수명 : 이경숙

학과 : 경제학과

이름 : 조신애

1. 수업 참여 후기

지난 1학기 동안, 대학교에 진학한 후, 첫 비대면 수업을 경험해보았습니다. 사실 코로나19가 시작되면서부터, 올해 3월까지 여전히 바뀐 환경에 적응하지 못했었습니다. 이 환경은 나에게만 주어진 것이 아닌, 전 세계 모든 사람에게 주어진 환경일 뿐더러 여기서 좌절하고, 불평만 늘어뜨려 놓아봤자, 누군가는 이 환경을 기회로 삼아 더 열심히 하고 있다는 것임을 인지하고 있었습니다. 그런데도, 점점 나태해져만 갔고, 이 환경에서 내가 공부를 잘할 수 있을까? 라는 걱정만 앞섰습니다. 그러나, 그런 제게 주어진 환경을 받아들이게 하고, 성실함까지 챙겨주어 극복하게 해준 수업이 있습니다. 교양수업인 ‘일본어 1’을 수강하면서, 코로나19 이후 변화한 환경에 갖고 있던 불만과 부정적인 인식을 전환하고, 나태함을 바로잡을 수 있었습니다. 지금부터, 수업을 통해 개인적으로 느꼈던 수업 후기와 더불어 교수님의 수업 방식을 소개해보도록 하겠습니다.

1) 녹화 강의 + Zoom 실시간 강의

이 수업에서 수요일은 실시간 강의 그리고 금요일은 녹화 강의로 진행되었습니다. 수요일 오전 10시 30분까지 Zoom을 통해 입장하면, 각자 채팅창에 본인의 이름을 적어 보냅니다. 이는 출석했다는 의미이기도 하고, 아침 수업의 분위기를 올리고자 하는 우리끼리의 방식이기도 했습니다. 과제 코멘트에다가도 항상 피드백을 달아주시지만, 교수님께서서는 본격적인 실시간 수업을 시작하시기 전에, 지난주 과제였던 작문과제 + 음독과제에 대한 피드백을 구두로 세세하게 진행하십니다. 히라가나, 가타카나 쓰는 법과 단어와 한자 읽는 법 등등 과제에서 학생들이 많이 놓쳤던 부분에 대한 실수를 줄이고자 강의에서 한 번 더 짚어주신 것 같습니다. 실시간 강의

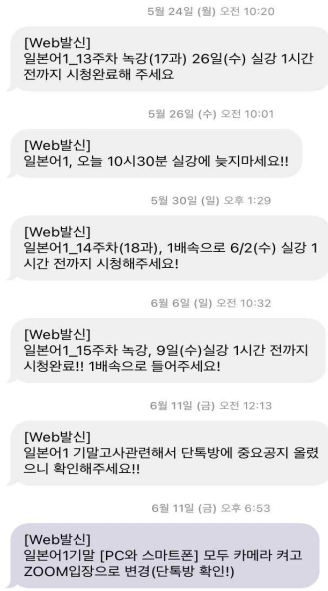
에서 수업하시는 내용은, 금요일에 녹화 강의로 들었던 내용을 바탕으로 하시기 때문에 녹화 강의를 꼭 성실히 들어야 했습니다. 녹화 강의 같은 경우는 성실하게 듣지 않아도 출석이 가능하므로, 자칫 잘못하면 바람직하지 않은 루틴이 형성될 수 있습니다. 하지만 실시간 강의를 듣다 보면, 내가 저번 주에 녹화 강의를 잘 들었는지 스스로 성찰하게 되어, 녹화 강의를 집중해서 듣는 습관을 기를 수 있었습니다. 또한 교과서 본문 음독 순서를 정하여, 해당 주치의 본문을 읽는 시간이 있습니다. 단순하게 읽고 끝나는 것이 아니라, 본문을 읽으면서 잘못된 발음이 있는지 봐주시기 때문에 자기가 맡은 것은 최선을 다해 준비한 후, 읽는 것이 중요했습니다. 따라서 실시간 강의와 녹화 강의를 적절히 같이 혼용하신 수업은 처음이었는데, 앞서 말한 여러 이유가 성실한 루틴을 지니게 해주어 굉장히 만족스러운 수업 방식이었습니다.

2) 작문과제 + 음독과제 : 복습, 충분한 피드백, 교수님과의 소통

교수님께서 내주시는 과제들의 정답은 모두 녹화 강의안에서 해결되기 때문에 따로 번역기를 돌리지 않아도 충분히 해결할 수 있습니다. 녹화 강의와 함께 올려주신 피피티 강의 자료를 보며 해당 주치의 내용을 자연스럽게 다시 한번 보게 됩니다. 과제를 해결하며, 교수님께서 중요하다고 강조하신 부분을 한 번 더 확인하게 되고, 심지어 내가 놓쳤던 부분을 발견하곤 했습니다. 교수님은 학생들이 비대면으로 나태해진다는 것을 알고 계시기에 복습을 할 수 있는 과제를 내주신 것 같았습니다. 음독과제 같은 경우에 혼자 녹음한 후, 제출할 때는 틀린 부분이 있는지 잘 인지하지 못했습니다. 하지만, 제출한 과제에 대한 피드백을 보고 난 후, 녹음을 다시 들어보면 어디가 틀렸는지 확 와닿았습니다. 이처럼 음독과제를 하면서, 완벽해질 때까지, 반복적으로 읽고, 피드백 해주신 것들을 바탕으로 잘못된 발음을 고치는데 집중하면서 읽게 되었습니다. 또한 앞서 말했듯이, 녹화 강의를 다시 보면서 과제를 한다고 했는데, 과제를 하면서 피피티에 오타를 발견할 때가 있습니다. 이를 교수님께 말씀드리면, 일본어1 단체 채팅방에 굉장히 빠르게 수정된 내용을 공지하시고, 피피티 강의 자료를 다시 올려주십니다. 비대면 이후로, 학생과 교수님과의 원활한 상호작용이 더욱 중요시되었다고 생각합니다. 비대면으로 인해 소통이 지체될 수 있는 가능성이 매우 큰데, 교수님께서 언제든 질문을 받아주시고, 빠르게 답해주셨습니다. 따라서 소통에 대한 문제점이 전혀 없었고, 감사드리는 마음뿐입니다.

2. 나만의 학습 노하우

1) 공지사항 꼼꼼하게 확인하기



1. 단체 채팅, 단체 문자: 질문&답변, 시험 주의사항, 출결 인정기준

일본어 1 수업에서는 문법, 단어, 한자 등등 암기가 요구되는 부분이 있습니다. 이러한 이유로, 녹화 강의를 제때 듣고, 정리할 시간을 마련하는 것이 학습에 도움이 된다고 느꼈습니다. 가끔, 녹화 강의 출석 인정 시간까지 최대한 미루고 싶은 생각이 들 때도 있었지만, 교수님께서 “녹화 강의 몇 시까지 인정되므로, 그때까지 들어주세요” 라는 단체 문자를 항상 보내주셨습니다. 비대면 이후로, 녹화 강의 출석 인정 날짜를 헛갈려서 실수로 결석한 경험이 있는 학생들이 많을 것입니다. 그러나 1학기 동안 교수님의 공지 덕분에, 일본어1 수업에서 결석 그리고 지각없이 모두 깔끔하게 출석이 완료되었습니다. 교수님께서 학점평가를 하실 때, 출결에 관해 강조를 해오셨습니다. 학습에 성실하게 임하는 태도를 갖추기 위해서는, 출석에 꼭 신경을 써야 합니다.

단체 채팅방 같은 경우에는 공지사항을 받는 것뿐만 아니라, 교수님께 적극적으로 질문할 수 있는 용도로도 사용되었습니다. 내가 생각하지 못한 질문들이 단체 채팅방에 올라오는 경우가 많습니다. 다른 학생들의 질문에 관심을 가지는 태도 또한 학습에 중요하다고 생각하기 때문에, 내가 질문한 것이 아니어도, 채팅방에 올라오는 질문과 그에 대한 교수님의 답변을 항상 확인했습니다. 비대면 시험으로 우려되는 부정행위를 방지하고자, 시험 주의사항들이 있었습니다. 시험 절차에 대한 학생들의 혼동을 막고자, 중간고사 그리고 기말고사 전에 시험 주의사항들을 정리하신 후에 공지해주십니다. 내가 학습한 내용을 시험에서 제대로 발휘하기 위해서는 공지사항을 꼼꼼하게 확인하여 시험에 임하는 것이 가장 중요하다고 생각합니다. 결과적으로 1학기 동안 아무런 문제 없이 원활하게 출석, 시험을 마무리할 수 있었습니다.

2) 긴장 풀기 & 점검하기 : 과제물 다시 풀어보기(*교과서, *강의자료 없이!)

저는 시험 보기 약 일주일 전부터, 걱정과 긴장을 많이 하는 편입니다. 공부에 집중해야 할 시간보다, 시험 걱정하는데 더 시간을 많이 쏟는 것 같았습니다. 과제물을 다시 풀어보며 내 실력도 점검할 겸, 긴장을 풀어야겠다 라는 생각을 했습니다. 미리 시험 보는 것처럼 연습하다 보면, 시험 시간에 긴장을 덜 하지 않을까 생각을 하여 타이머를 맞춰놓고 과제물을 풀어갔습니다. 한번 풀어본 과제여서 답이 기억

