

정보과학

1. 성격 및 목표

가. 성격

‘정보(Informatics)’과는 인공지능으로 정의되는 사회에서 데이터와 정보로 인한 디지털 세상의 변화를 인식하고, 정보의 사회적 가치를 탐구하며, 정보를 처리하는 다양한 원리와 기술에 기반한 컴퓨팅 사고력을 바탕으로 실생활 및 다양한 학문 분야의 문제를 해결하는 능력과 태도를 기르는 교과이다. ‘정보’는 디지털 대전환 시대의 국가사회적 요구에 부응하여, 컴퓨팅을 활용한 문제 해결을 위해 사회 구성원이 갖추어야 할 필수 역량을 제공한다. ‘정보’의 학문적 기저는 컴퓨터에서 처리되는 데이터와 정보의 원리, 컴퓨팅 시스템을 설계하고 구현하는 기술과 방법, 정보를 다루는 인간 사회에 대한 이해 등을 포괄하고 있다. 즉, ‘정보’는 컴퓨터과학뿐 아니라 데이터 과학, 인공지능, 정보기술, 정보시스템, 소프트웨어 공학 등의 분야를 포괄하는 정보학에 대한 기본 개념과 원리를 기반으로 다양한 학문 분야와 미래 사회의 문제를 해결하는 데 도움이 되는 지식과 기술을 함양한다. 교과의 이러한 특성은 사회 각 분야에서 요구되는 소프트웨어와 인공지능에 대한 기본 소양을 갖추고, 공학뿐만 아니라 자연과학, 인문·사회과학, 예술과 체육 등 다양한 학문 분야에서 문제를 창의적으로 해결하는 인재 양성에 도움을 준다.

‘정보과학’은 컴퓨터과학과 소프트웨어 공학에 대한 깊이 있는 학문적 이해를 바탕으로 디지털 사회의 복잡한 문제를 이해하고, 미래 사회에 발생 가능한 문제를 해결하는 데 도움이 되는 문제 분석, 구조화, 해결을 위한 알고리즘 설계가 프로그래밍을 통해 구체화되는 과정을 경험할 수 있게 한다. 소프트웨어에 대한 깊이 있는 학습을 기반으로 대학의 소프트웨어 관련 다양한 전공과 연계되는 기초 지식을 습득하고, 컴퓨팅의 기본 개념과 원리, 컴퓨팅 시스템 등을 활용한 심도 있는 문제 해결 역량을 함양한다. 이를 통해 실세계 및 타 학문 분야의 문제를 융합적이고 창의적으로 해결할 수 있는 역량을 갖춘 디지털 민주시민으로 성장하게 된다.

나. 목표

고등학교 ‘정보과학’은 컴퓨터과학의 기본 개념과 원리, 컴퓨팅 기술을 기반으로 디지털 사회를 살아가는 데 필요한 기본 역량을 함양하고, 실생활에서 발생하는 문제와 다양한 학문 분야의 문제를 융합적으로 해결하기 위한 정보과학적 방법론을 습득하여 해결할 수 있는 능력을 기르는 데 중점을 둔다.

- (1) 문제를 해결하기 위한 프로그래밍 과정을 통해 정확하고 효율적으로 작동하는 소프트웨어를 개발할 수 있는 능력을 기른다.
- (2) 데이터의 처리를 위해 선행되어야 할 데이터의 구조에 대해 파악하고 효율적인 문제 해결을 위한 데이터 구조화의 가치를 파악하는 태도를 기른다.
- (3) 실생활 및 다양한 학문 분야의 문제를 해결하는 과정을 통해 알고리즘 설계 및 구현 능력을 기른다.
- (4) 컴퓨터과학의 기본 개념과 원리에 기반하여 타 학문 분야의 문제를 해결하기 위한 융합적 설계, 새로운 방법론 활용 등의 능력을 기른다.

2. 내용 체계 및 성취기준

가. 내용 체계

(1) 프로그래밍

핵심 아이디어	<ul style="list-style-type: none"> • 프로그래밍을 통한 자동화는 다양한 학문 분야의 문제를 해결하는 데 도움을 준다. • 효율적인 프로그래밍을 가능하게 하는 함수는 프로그램의 간결화, 재사용성 측면에서 활용된다. • 프로그램 개발은 협력이 필요하며, 공유하는 문화를 통해 더 좋은 프로그램이 개발된다.
범주	구분 내용 요소
지식·이해	<ul style="list-style-type: none"> • 함수 정의와 호출 • 재귀관계와 재귀함수
과정·기능	<ul style="list-style-type: none"> • 매개변수를 활용한 함수 프로그램 작성하기 • 문제 해결을 위한 재귀관계 파악 및 재귀함수 구현하기
가치·태도	<ul style="list-style-type: none"> • 재귀구조의 가치를 이해하고 적극적으로 활용하는 자세 • 문제 해결에 새로운 아이디어를 적용하는 자세

(2) 데이터 구조

핵심 아이디어	<ul style="list-style-type: none"> • 문제에 따라 해결에 적합한 데이터 구조를 선택하는 것이 중요하다. • 데이터들의 관계를 파악하고 특성에 맞도록 구조화하는 것은 문제를 효율적으로 해결하는 데 도움을 준다.
범주	구분 내용 요소
지식·이해	<ul style="list-style-type: none"> • 스택과 큐 • 트리와 그래프
과정·기능	<ul style="list-style-type: none"> • 순차적인 데이터 구조를 이용하여 스택, 큐 구현하기 • 스택, 큐를 활용하여 해결할 수 있는 문제 탐색하기 • 인접행렬과 인접리스트로 트리, 그래프 구현하기 • 트리, 그래프를 활용하여 해결할 수 있는 문제 탐색하기
가치·태도	<ul style="list-style-type: none"> • 데이터 구조화를 통해 문제를 해결하는 태도 • 효율적 문제 해결을 위한 데이터 구조화의 중요성 인식

(3) 알고리즘

핵심 아이디어	<ul style="list-style-type: none"> • 자동화를 고려해 설계된 알고리즘은 컴퓨터가 이해할 수 있는 언어로 구현되어 문제 해결에 도움을 준다. • 문제를 효율적으로 해결하기 위해서는 적합한 알고리즘 설계 전략을 선택하는 것이 중요하다. • 알고리즘 효율을 개선하기 위해 알고리즘의 성능을 정량적으로 분석한다.
범주	구분
	내용 요소
지식·이해	<ul style="list-style-type: none"> • 탐색기반 알고리즘 • 관계기반 알고리즘 • 알고리즘 복잡도
과정·기능	<ul style="list-style-type: none"> • 문제 상태 공간을 탐색하고, 알고리즘 설계하기 • 문제를 분해하고, 모델링하기 • 빅오 표기법으로 알고리즘 성능 비교하기
가치·태도	<ul style="list-style-type: none"> • 문제 해결 상황에 적절한 알고리즘을 적용하는 유연한 태도 • 정량적인 분석을 통해 알고리즘의 성능을 객관적으로 평가하는 자세

(4) 정보과학 프로젝트

핵심 아이디어	<ul style="list-style-type: none"> • 정보과학 프로젝트를 수행하는 것은 다양한 학문 분야의 문제를 컴퓨팅 관점에서 해결하는 데 필요하다. • 프로젝트를 수행할 때, 오픈소스의 활용, 수행된 프로젝트의 공유, 다양한 사람들과의 협업 등은 더 좋은 사회를 만드는 데 도움을 준다.
범주	구분
	내용 요소
지식·이해	<ul style="list-style-type: none"> • 문제 발견 • 프로젝트 설계 • 오픈소스와 공유 • 테스트와 디버깅
과정·기능	<ul style="list-style-type: none"> • 다양한 학문 분야에서 컴퓨터과학의 원리를 바탕으로 해결 가능한 문제 탐색하기 • 문제를 분석하고, 문제 해결을 위한 프로젝트 설계하기 • 공유된 오픈소스를 활용하여 문제를 해결하는 프로그램 작성하기 • 프로젝트 산출물을 평가하고 공유하기
가치·태도	<ul style="list-style-type: none"> • 협력적으로 문제를 해결하고 공유하는 태도 • 프로젝트를 수행하는 과정에서 윤리 문제 등 사회적 영향 인식

나. 성취기준

(1) 프로그래밍

[12정과01-01] 함수 정의와 호출의 원리를 이해하고, 매개변수를 활용한 함수 프로그램을 작성한다.

[12정과01-02] 주어진 문제에서 적용할 수 있는 재귀관계를 파악하고, 재귀구조를 활용하여 프로그램을 작성한다.

[12정과01-03] 반복구조와 재귀구조를 활용하여 문제를 해결하고, 두 방식의 차이를 비교분석한다.

(가) 성취기준 해설

- [12정과01-02] 많은 알고리즘의 코드가 재귀적으로 구성되어 있고 재귀구조를 통해 높은 수준의 생산성을 기대할 수 있다. 문제 내에서 재귀관계를 파악할 때는 수학적 귀납법을 활용할 수 있음을 이해하고, 문제 해결책을 재귀구조를 활용하여 설계하고 프로그래밍할 수 있어야 한다. 또한, 이후의 학습 내용이 많은 부분 재귀구조와 관련되어 있음을 판단할 수 있어야 한다.
- [12정과01-03] 재귀구조와 반복구조는 문제 해결책을 설계하는 방법이나 코드의 표현에 있어서 서로 다른 지향점을 갖는다. 특정한 문제를 해결할 때 재귀구조를 사용하여 프로그래밍하는 것과 반복구조를 사용하여 프로그래밍하는 방식의 차이를 구체적으로 비교하여 설명할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 수학적 귀납법을 적용할 때 주어진 명제를 수학적으로 증명하는 과정을 설명하는 방법보다는 귀납 가정 부분이 재귀함수 호출과 어떤 관계가 있는지를 다양한 예를 통해서 이해할 수 있도록 하여 실제 문제를 해결할 때 재귀구조를 활용할 수 있음을 인식하도록 한다.
- 최소 성취수준을 보장하기 위해 주어진 문제로부터 재귀관계를 파악하지 못하더라도 반복구조로 작성된 프로그램을 재귀함수로 변환하여 구현하는 방식의 활동을 제공하도록 한다.

(2) 데이터 구조

[12정과02-01] 스택과 큐의 원리를 이해하고, 순차적인 데이터 구조를 이용하여 스택과 큐를 구현한다.

[12정과02-02] 스택, 큐를 활용하여 문제를 효율적으로 해결하는 프로그램을 작성한다.

[12정과02-03] 인접행렬과 인접리스트를 활용하여 트리와 그래프를 구현한다.

[12정과02-04] 트리, 그래프를 활용하여 문제를 효율적으로 해결하는 프로그램을 작성한다.

(가) 성취기준 해설

- [12정과02-01] 선입선출, 후입선출 구조를 활용하면 복잡한 문제를 간단하게 해결할 수 있음을 이해할 수 있어야 한다. 프로그래밍 언어의 특성에 따라 배열이나 리스트를 활용하여 대표적인 선입선출, 후입선출 구조인 스택과 큐를 구현하고 문제를 해결하는 데 활용할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 스택과 큐를 배열이나 리스트로 구현하는 활동을 통하여 원리를 이해할 수 있도록 하고, 이후 진행되는 문제 해결 활동에서는 프로그래밍 언어별로 활용 가능한 스택과 큐의 라이브러리를 활용하여 구현하는 방법도 활용하도록 한다.
- 인접리스트는 트리와 그래프를 만드는 도구로서의 활용이 목적이므로 구조체와 포인터 등으로 직접 구현하는 방법보다는 프로그래밍 언어별로 제공하는 벡터(vector)와 같은 라이브러리를 이용하여 구현하는 방법도 활용하도록 한다.
- 데이터 구조의 원리를 이해하고 구현하는 과정이 궁극적으로는 문제를 효율적으로 해결하는 방법임을 이해하고 학습을 진행하는 것이 필요하다. 따라서 데이터 구조를 활용한 실생활의 문제 해결 활동과 예제를 제공하는 방식으로 교수·학습을 구성하고 해당 맥락에서 데이터 구조를 활용하는 능력을 향상하도록 한다.

(3) 알고리즘

[12정과03-01] 문제를 분석하여 상태 공간으로 구조화하고, 상태 공간을 모두 탐색하여 문제를 해결하는 알고리즘을 설계한다.

[12정과03-02] 문제의 상태 공간 일부를 배제하는 알고리즘을 설계하고, 기존의 탐색기반 알고리즘과 비교·분석한다.

[12정과03-03] 전체 문제를 부분 문제로 분해하고, 전체 문제와 부분 문제의 관계를 발견한다.

[12정과03-04] 부분 문제를 해결한 결과를 이용하여 전체 문제를 해결하는 알고리즘을 설계한다.

[12정과03-05] 부분 문제가 중복으로 적용되지 않는 알고리즘을 설계하고 기존의 관계기반 알고리즘과 비교·분석한다.

[12정과03-06] 알고리즘의 복잡도를 빅오 표기법으로 표현하고 다른 알고리즘과 비교한다.

(가) 성취기준 해설

- [12정과03-02] 상태 공간 일부를 배제하고도 정확한 답을 구할 수 있는 분기한정법과 탐욕법 등의 알고리즘 설계 방법이 상태 공간을 배제하는 원리를 설명하고, 이를 활용하여 문제를

해결할 수 있는 알고리즘을 설계할 수 있어야 한다.

- [12정과03-04] 퀵 정렬, 합병 정렬 등을 활용하여 분할정복법이 문제를 해결하는 원리를 이해하고, 분할정복법을 활용하여 해결할 수 있는 문제를 탐색하여 해결 알고리즘을 설계할 수 있어야 한다.
- [12정과03-05] 배열 등을 활용하여 중복으로 계산되는 부분 문제들의 답을 저장하여 효율을 높이는 동적계획법의 원리를 설명하고, 이를 활용하여 문제를 해결하는 알고리즘을 설계할 수 있어야 한다. 동적계획법과 분할정복법을 기반으로 한 알고리즘 간의 차이를 여러 측면에서 설명할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 탐색으로 알고리즘을 설계하는 방법은 인공지능 분야의 탐색과 추론영역과 연계됨을 인식하고 교수·학습을 구성할 때 관련된 용어가 명확하게 사용되는지 유의하도록 한다.
- 알고리즘 영역에서 제시되는 구체적인 알고리즘은 정보학의 문제 해결 과정에서 중요하게 활용되고 있음을 이해할 수 있도록 실제적인 예시를 통해 교수·학습을 구성하도록 한다. 이때 실제 알고리즘이 적용된 프로그램이나 시각적인 자료를 통해 학습자가 더욱 직관적으로 효율성을 인식하도록 구성하는 방식도 고려할 수 있다.
- 알고리즘을 적용한 프로그램의 평가 시 알고리즘의 효과성과 효율성을 종합적으로 평가하도록 한다. 특히 다양한 프로그램 환경이나 데이터를 활용하여 설계한 알고리즘이 동일하게 효율적으로 작동하는지를 평가하여 학습 내용이 확장·전이되었는지를 확인하도록 한다.
- 최소 성취수준을 보장하기 위하여 알고리즘을 경험할 수 있는 다양한 시뮬레이션 자료를 제공하고, 프로그래밍 언어가 제공하는 라이브러리 활용 방법을 안내하여 최소한의 알고리즘 적용 활동을 진행할 수 있도록 한다.

(4) 정보과학 프로젝트

[12정과04-01] 다양한 학문 분야에서 컴퓨터과학의 기본 개념과 원리를 바탕으로 해결할 수 있는 실생활의 문제를 발견한다.

[12정과04-02] 발견한 문제를 분석하고, 문제를 해결하기 위한 프로젝트 수행 과정을 설계한다.

[12정과04-03] 다양한 오픈소스 및 라이브러리를 활용하여 협력적으로 문제를 해결하기 위한 프로그램을 작성한다.

[12정과04-04] 완성도 높은 프로그램을 구현하기 위하여 테스트와 디버깅 과정을 통해 검증한다.

[12정과04-05] 프로젝트의 산출물을 기능적, 윤리적 관점에서 평가하고 결과를 공유한다.

(가) 성취기준 해설

- [12정과04-02] 프로젝트 수행 과정은 프로그래밍을 활용한 문제 해결 절차 즉, 문제 분석

및 구조화, 효율적으로 해결할 수 있는 알고리즘 설계, 프로그래밍을 통한 자동화, 그리고 검증하는 단계까지 포함하고 있음을 이해하고, 절차에 맞게 문제 해결 과정을 설계할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 실생활의 다양한 문제를 발견하는 활동에서는 환경, 생태, 자연의 위기 등 생태 관련 주제도 포함될 수 있음을 안내하고, 이를 효율적으로 해결하는 것은 에너지 절약을 비롯한 환경을 보호하는 것과 관계가 있음을 인식할 수 있도록 한다.
- 프로젝트 활동은 주어진 문제를 해결하기 위한 알고리즘 설계 및 프로그래밍 과정을 포함하도록 교수·학습을 구성하며 주제 선정, 프로그래밍 언어 선정 등의 절차는 학습자 간의 협의를 통하여 민주적 절차로 이루어질 수 있도록 한다.
- 프로젝트를 해결하는 과정은 가능한 모듈별로 진행하고, 자신이 맡은 역할에 충실한 방식으로 협업이 원활하게 진행되도록 교사가 중재하도록 한다. 프로그래밍 과정에서는 세부적인 기능들을 모두 구현하는 것보다는 공개된 프로그램 코드나 프로그래밍 언어에서 제공하는 라이브러리를 적극적으로 활용하여, 목표로 하는 기능을 충분히 구현하고 프로젝트 과제를 해결할 수 있도록 한다.
- 프로젝트 주제에 따라 목적에 맞는 소프트웨어를 제작하는 활동이나 특정 문제를 해결하는 알고리즘을 설계하는 등 다양한 수준의 활동이 가능함을 이해하고 학생들의 역량과 환경에 따라 적절한 형태로 운영할 수 있도록 다양한 사례를 제시하고 활용하도록 한다.

3. 교수·학습 및 평가

가. 교수·학습

(1) 교수·학습의 방향

(가) 실제적인 삶의 맥락에서 컴퓨팅을 통해 문제를 해결하도록 하는 학습 과제를 제시하여 학습자가 과제를 스스로 해결하는 과정에서 자연스럽게 컴퓨팅 사고력, 디지털 문화 소양, 인공지능 소양을 함양할 수 있도록 지도한다.

(나) 학습자의 흥미와 다양성을 고려하여 학습 소재, 학습 환경 및 학습 과정에 대한 선택의 기회를 제공하고, 교수·학습의 설계 과정에 학습자 참여 기회를 증진하는 등 학습자 맞춤형 교수·학습을 통해 역량 함양을 위한 깊이 있는 학습 지도 방안을 구성한다.

(다) 정보과학 과목의 지식·이해, 과정·기능을 활용하여 민주시민교육, 생태전환 교육 등 현 시대가 당면한 여러 사회문제와 더불어 지속가능발전 등의 범교과 주제를 교수·학습 과제로 제시하여 주도성 있는 문제 해결 경험을 제공한다.

(라) ‘정보과학’ 과목의 교수·학습은 컴퓨터과학 분야의 깊이 있는 이해를 바탕으로 문제를 효율적으로 해결하는 역량을 향상하는 데 중점을 둔다. 또한 ‘정보과학’ 과목 내의 내용 영역, 다른 교과 및 비교과 활동과의 융합을 통해 정보 관련 역량의 확장을 꾀하고 학생의 역량이 다양한 분야에 전이될 수 있도록 한다.

(2) 교수·학습 방법

(가) 컴퓨팅 사고력 및 인공지능 소양을 함양하기 위해 문제기반학습, 프로젝트 기반학습, 디자인기반학습, 탐구학습 등 각 영역의 핵심 아이디어를 습득하는 데 적절한 교수·학습 방법을 선택하여 활용한다.

(나) 학습자 개인별로 학습하는 속도가 다양할 수 있음을 고려하고, 최소 성취수준을 보장할 수 있도록 학습관리시스템(LMS)을 활용하여 온라인 학습자료를 제작 및 제공함으로써 학습 격차를 최소화하도록 노력한다.

(다) 개인적인 발전과 함께 협력적인 문제 해결력을 기르기 위해 프로젝트 학습 시에는 모둠 프로젝트, 짝 프로그래밍, 집단 탐구(GI) 등 목표 달성을 위한 적절한 협력적 교수·학습 방법을 선택하여 활용한다.

(라) 디지털 교육 환경에 적응할 수 있도록 온·오프라인 연계 수업, 다양한 디지털 도구의 활용 등을 통해 디지털 도구에 대한 인지적 부담은 최소화하고, 활용에 대한 경험은 높일 수 있도록 수업을 구성한다.

(마) ‘정보과학’ 과목에 포함된 내용 영역과 연계된 진로 및 직업을 안내하고 학생이 자신의 진로를 탐색할 수 있도록 교수·학습 과정을 구성한다.

나. 평가

(1) 평가의 방향

(가) 평가 항목은 컴퓨팅 사고력, 디지털 문화 소양, 인공지능 소양의 하위 요소를 기반으로 구체화한다.

(나) 평가 내용은 지식·이해뿐만 아니라, 과정·기능, 가치·태도의 측면 등을 다면적으로 반영하고 과정을 중시하는 평가를 통해 학생의 성장과 발달을 돕는 평가를 실현한다.

(다) 구체적인 평가 루브릭을 학생과 함께 구성하는 과정을 통해 학생이 자신의 학습 수준을 파악하고 스스로 학습을 성찰할 수 있는 기회를 제공하여, 적극적이고 능동적인 학습이 이루어지도록 한다.

(라) 단순하고 지엽적인 지식의 평가보다는 문제를 해결하는 과정을 통합적으로 관찰하고 평가할 수 있는 계획을 수립한다.

(마) 개념적이거나 기능적으로 명확하게 파악할 수 있는 부분은 정량적 평가를 실시하고, 결과물의 품질이나 심미적 부분을 평가할 때는 정성적 평가를 실시한다.

(2) 평가 방법

(가) 성취기준을 분석하고 재구성하여 지필평가에 국한하지 않고, 학생의 성장에 기여할 수 있는 평가 포트폴리오를 계획한다. 예를 들면, 관찰 평가, 서술형평가, 수행평가 등을 활용하거나, 자기 평가, 동료 평가 등과 같은 다면적 평가를 실행한다.

(나) 평가 내용이나 방법에 따라 다양한 디지털 도구(프로그램 자동 평가시스템(online judge 등), 학습관리시스템(LMS) 등)를 활용할 수 있으며, 평가 이전에 학생이 디지털 도구를 다룰 수 있도록 교육하여 평가의 불이익이 없도록 계획한다.

(다) 알고리즘을 평가할 때 가능하면 프로그램 자동 평가시스템을 활용하여 컴퓨팅 시스템에서 실제로 구현되는 결과를 평가하도록 하며, 해당 방법을 활용하기 어려운 경우 대표성을 지니는 데이터를 선정하여 평가 문항과 루브릭을 구성하고, 지필평가나 수행평가의 형태로 평가를 진행하여 공정한 평가가 되도록 한다.