

# 소프트웨어공학 오리엔테이션

## 과정 인사말

안녕하십니까?

앞으로 한 학기 동안 소프트웨어공학을 통해 여러분과 만나게 될 전우천 교수입니다.

먼저 소프트웨어공학 과목에 관심을 가져주셔서 감사 드리고 여러분을 환영합니다.

소프트웨어공학 과목은 전산학의 주요과목입니다.

흔히 전산학의 5개 주요과목으로서

전자계산기구조, 운영체제, 소프트웨어공학, 정보통신, 데이터베이스(자료구조 포함)입니다.

# 소프트웨어공학 오리엔테이션

## 과정 인사말

소프트웨어공학은 한마디로 소프트웨어를 제작할 때  
처음부터 마지막까지 전 과정을 글자 그대로  
공학적으로 처리하기 위한 과목입니다.

즉 소프트웨어를 제작할 때의 첫 과정인 사용자  
요구분석으로부터 시작해서, 소프트웨어 설계, 개발,  
테스트를 거쳐 제품을 완료하여 발주자에게  
인도한 후 이후에 발생할 수 있는 다양한 유지보수까지  
포함하는 전 과정에 최대한 공학적으로 접근하여  
최소한의 비용과 시간으로 최대의 효과를 거두기 위한  
과목입니다.

# 소프트웨어공학 오리엔테이션

## 과정 인사말

소프트웨어공학과목은 다른 전산학과목에 비해서  
수학적 배경지식이 비교적 덜 요구되지만  
과목을 이해하기 위해서는 기본적으로는 프로그래밍  
경험, 더욱 나아가서는 대규모 프로그래밍 프로젝트  
경험이 있어야 과목을 이해하는 데 도움이 됩니다.

앞으로 본 과목을 충실히 이행해나가고 무엇보다도  
흥미를 가지고 접근하는 자세를 가지시길 바랍니다.

감사합니다.

# 소프트웨어공학 오리엔테이션

## 교수 소개

- 현) 서울교육대학교 컴퓨터교육과 교수
- 현) 서울교육대학교 영재교육원 정보영재반 교수
- 현) 한국인터넷정보학회 학회지 편집위원장
- 현) 한국창의정보문화학회 논문지 편집위원장

# 소프트웨어공학 오리엔테이션

## 교수 소개

- 🔍 전) 한국정보교육학회 총무이사
- 🔍 전) 서울교육대학교 정보전산원장
- 🔍 전) 한국정보화진흥원 정보통신윤리 전문강사
- 🔍 오클라호마주립대 전산학 박사

# 소프트웨어공학 오리엔테이션

## 과목 소개

- 🔍 본 과목은 소프트웨어를 설계하고 제작하고 관리하고 유지 보수하는 전 과정에 대해서 학습함
- 🔍 소프트웨어가 점차 대형화, 구조화, 및 다변화가 되면서 공학적으로 접근하여 관리할 필요성이 대두됨
- 🔍 먼저 소프트웨어 개발 프로젝트, 프로젝트 관리, 개발 프로세스에 대해서 전반적으로 학습함
- 🔍 이를 기반으로 사용자요구사항분석, 설계원리 (소프트웨어 아키텍처, 데이터베이스, 사용자인터페이스), 구현, 테스트, 품질관리, 위험관리, 형상 관리, 유지보수에 대해서 체계적으로 학습

# 소프트웨어공학 오리엔테이션

## 강의 구성

- 🔍 4차 산업혁명시대에서 소프트웨어의 중요성 및 소프트웨어공학의 필요성 이해
- 🔍 소프트웨어 개발 프로젝트, 프로젝트 관리, 소프트웨어 개발 방법론에 대해서 학습
- 🔍 구조적 분석기법 및 객체지향적 분석 학습
- 🔍 소프트웨어 설계의 기본방법론을 비롯하여 소프트웨어 아키텍처 설계, 데이터베이스 설계, 사용자인터페이스 설계를 학습함
- 🔍 구현 및 테스트 방법론, 품질관리, 위험관리, 형상관리, 프로젝트 종료 및 유지보수에 대해서 체계적으로 학습

# 소프트웨어공학 오리엔테이션

## 강의 활용 Tip

- 🔍 인터넷과 잡지 등을 통하여 다양한 소프트웨어 프로젝트의 성공 및 실패 사례 분석이 필요함
- 🔍 3세대 고급언어 또는 4세대 고급언어를 하나 선정하여 체계적으로 학습하여 소프트웨어를 개발할 기본능력을 갖추 필요 있음
- 🔍 가능하면 3~4명의 팀을 구성하여 실제로 소프트웨어 개발하여 완료하는 계획을 세우고 실천할 필요 있음

1

## 소프트웨어의 정의

# 01 소프트웨어의 정의

## 1 소프트웨어공학이 무엇인가요?

 소프트웨어의 위기(Software Crisis)를 극복하기 위하여

- 소프트웨어 개발에 공학적 패러다임을 적용하자는 취지로 시작

 소프트웨어의 위기(Software Crisis)

- 1960년대 소프트웨어가 대단한 각광을 받으면서 폭발적인 수요에 공급이 따라가지 못하면서 실패한 프로젝트가 엄청나게 발생

# 01 소프트웨어의 정의

## 1 소프트웨어공학이 무엇인가요?



1968년 NATO의 한 컨퍼런스 회의에서

- 의장을 맡았던 “프리드리히 바우어”에 의해 제안



“Software Engineering”이라는 용어가 처음으로 제시됨

# 01 소프트웨어의 정의

## 1 소프트웨어공학이 무엇인가요?

### 소프트웨어공학에 대한 정의 1

- 기계에서 효율적으로 작동하는 신뢰성 있는 소프트웨어를 경제적으로 획득하기 위해 적절한 공학적 원리를 수립하여 활용하는 것(바우어)
- “Establishment and use of sound engineering principles to economically obtain software that is reliable and works on real machines efficiently” (바우어)

# 01 소프트웨어의 정의

## 1 소프트웨어공학이 무엇인가요?

### 소프트웨어공학에 대한 정의 2

- 소프트웨어의 개발과 운용, 유지보수에 대한 체계적(Systematic)이며, 훈련된(Disciplined) 계량적(Quantifiable) 접근 방식의 적용 (IEEE)
- “the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” (IEEE)

# 01 소프트웨어의 정의

## 2 소프트웨어에 대한 이해

### 솜머빌(Sommerville)

소프트웨어는 단순한 프로그램 뿐만 아니라  
프로그램이 올바르게 작동하도록 하는데 필요한  
관련된 문서 및 설치 데이터를 의미




#### 소프트웨어 구성요소

- 프로그램
- 문서
- 데이터

# 01 소프트웨어의 정의

## 2 소프트웨어에 대한 이해

### 프레스만(Pressman)

-  실행되면서 원하는 기능이나 함수, 성능을 제공하는 명령어들(컴퓨터 프로그램)
-  프로그램이 데이터를 적절하게 처리할 수 있게 해 주는 자료구조
-  프로그램의 사용이나 운영을 나타내는 하드카피나 가상의 형태의 문서

# 01 소프트웨어의 정의

## 2 소프트웨어에 대한 이해

### 프레스만(Pressman)

#### 소프트웨어 구성요소

- 프로그램
- 자료구조
- 문서

# 01 소프트웨어의 정의

## 2 소프트웨어에 대한 이해

### 자료구조와 자료, 과연 소프트웨어인가?

- 백과사전 소프트웨어에서 콘텐츠는 자료인가?  
콘텐츠가 없다면 백과사전이라고 할 수 있을까?
- 게임 소프트웨어에서 각종 아이템과 영웅이미지,  
소프트웨어인가? 영웅이미지가 나타나지  
않는다면 제대로 된 소프트웨어라고  
할 수 있을까?
- 자료구조가 없다면 각종 데이터를 어떻게  
저장할 것인가? 소프트웨어 내부 코드들은  
어떻게 관리되어야 할 것인가?

# 01 소프트웨어의 정의

## 2 소프트웨어에 대한 이해

 문서도 소프트웨어인가?

- 온라인 매뉴얼이 없는 소프트웨어의 사용은?
- 사용자 교육, 유지보수를 위한 개발자 교육은 무엇으로 할 것인가?
- 소프트웨어 개발결과를 계약하고자 한다면 결과에 대한 증빙은 무엇으로 어떻게 하는 것이 좋을까?

# 01 소프트웨어의 정의

## 3 소프트웨어의 특징과 분류



### 소프트웨어의 특징

#### 비가시성

- 완성되기 전까지 전체를 파악하기 힘들

#### 변경성

- 완성되기 전까지 계속 바뀌며, 완성된 이후에도 변경됨

#### 복제성

- 무한 복제 가능, 닳거나 없어지지 않음

#### 복잡성

- 개발과정이 대단히 복잡함

# 01 소프트웨어의 정의

## 3 소프트웨어의 특징과 분류

### 소프트웨어의 분류

분류	내용
시스템 소프트웨어 (System Software)	다른 프로그램의 수행을 지원해주기 위해 만들어진 소프트웨어
애플리케이션 소프트웨어 (Application Software)	특수한 업무상의 요구를 해결해주며, 비즈니스 처리 또는 관리/기술 측면에서의 의사결정을 쉽게 할 수 있도록 사업상 또는 기술적 데이터를 처리하는 소프트웨어

# 01 소프트웨어의 정의

## 3 소프트웨어의 특징과 분류

### 소프트웨어의 분류

분류	내용
공학/과학 소프트웨어 (Engineering/Science Software)	“수 처리” 알고리즘이 특징이며, 공학 및 과학적 연구를 위해, 실시간 처리가 가능한 소프트웨어
임베디드 소프트웨어 (Embedded Software)	생산품이나 시스템에 내장되어 있으면서 사용자나 시스템 자체를 위한 특징이나 기능을 구현하는데 사용

# 01 소프트웨어의 정의

## 3 소프트웨어의 특징과 분류



### 소프트웨어의 분류

분류	내용
웹 응용시스템 소프트웨어 (Web Application)	인터넷을 위해 개발된 하이퍼텍스트 이거나 , 웹으로 가동되는 응용시스템을 위한 소프트웨어
인공지능 소프트웨어 (Artificial Intelligence Software)	계산이나 일반적인 알고리즘으로 분석할 수 없는 복잡한 문제를 해결하기 위해 비수치적 알고리즘을 사용

# 01 소프트웨어의 정의

## 3 소프트웨어의 특징과 분류



### 소프트웨어의 분류

분류	내용
생산라인 소프트웨어 (Product-Line Software)	제조공정과정에 필요한 소프트웨어

2

## 소프트웨어의 탄생과 폐기

## 02 소프트웨어의 탄생과 폐기

### 1 소프트웨어의 탄생과 폐기과정

사람처럼 소프트웨어도 태어나고 사라짐(Life Cycle)

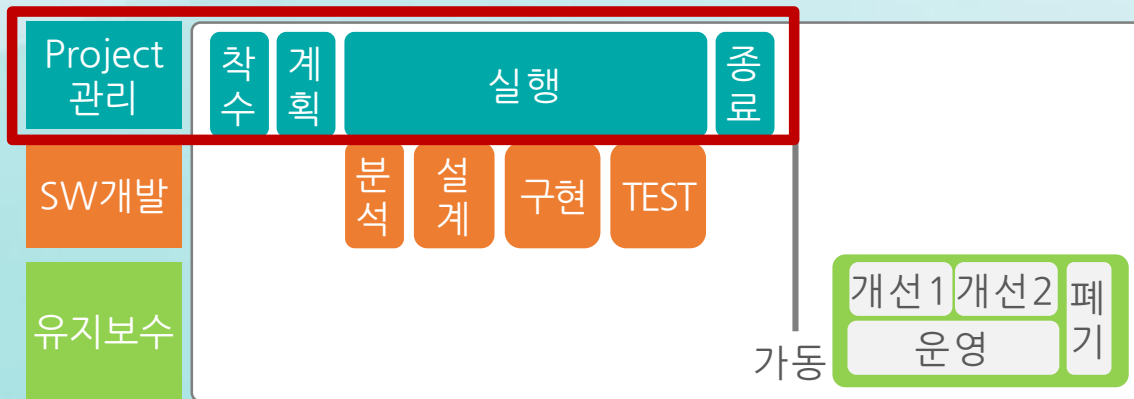


## 02 소프트웨어의 탄생과 폐기

### 2 프로젝트 관리

🔍 소프트웨어를 개발하기 위한 아이디어  
구상 및 팀 구성 등

➡ 목적(필요성), 예산, 기간, 팀원 및 역할 등



## 02 소프트웨어의 탄생과 폐기

### 3 소프트웨어 개발

🔍 소프트웨어 개발 라이프사이클(SDLC, Software Development Life Cycle)에 따라 진행



## 02 소프트웨어의 탄생과 폐기

### 4 유지보수

🔍 실제로 소프트웨어가 전기를 먹고 사람들의 도움을 받으며 살아 움직이는 과정, 폐기에 의해 세상을 하직



## 02 소프트웨어의 탄생과 폐기

### 5 유지보수의 종류

종류	변경의 강도	내용
완전 유지보수 (Perfective Maintenance)	상	새로운 요구사항을 추가하거나 시스템의 구조와 성능을 개선하여 시스템을 완전하게 만드는 목적으로 수행
수정 유지보수 (Corrective Maintenance)	중	요구사항의 오류나 설계 및 구현 상의 오류를 개선할 목적으로 수행함

## 02 소프트웨어의 탄생과 폐기

### 5 유지보수의 종류


종류	변경의 강도	내용
적응 유지보수 (Adaptive Maintenance)	하	시스템의 플랫폼 변경과 같은 새로운 환경으로의 적응을 목적으로 수행
예방 유지보수 (Preventive Maintenance)	하	시스템의 잠재적인 결함을 사전에 방지하기 위한 목적으로 수행

### 3 다양한 소프트웨어 도입 프로젝트

## 03 다양한 소프트웨어 도입 프로젝트

### 1 인하우스 및 SI 프로젝트

#### 인하우스(In house) 개발 프로젝트


 조직 내부 자체 인력으로 소프트웨어를 개발

- 소규모 프로젝트 추진 시
- 프로젝트 계약과정이 불필요
- 일부 필요 시 개발자 혹은 전문가를 초빙

## 03 다양한 소프트웨어 도입 프로젝트

### 1 인하우스 및 SI 프로젝트


#### SI(System Integration) 프로젝트

 대규모 프로젝트 혹은 위험한 프로젝트의 경우 외부 전문 SI 업체에 맡겨 프로젝트 수행

- REP 제시
- 업체를 선정하기 위한 과정이 필요
- 선정된 업체와 프로젝트에 대한 계약 필요

## 03 다양한 소프트웨어 도입 프로젝트

### 2 패키지 도입

 조직 레벨에서 활용할 패키지의 도입

- ERP(SAP 혹은 Oracle)을 도입하는 프로젝트가 안 예임
- 이미 만들어진 기능을 대부분 활용 혹은 커스터마이징을 통해 기능을 보완
- 전혀 변경이 없이 활용하기도 함