

모의 고사 2회 답안 및 해설

[Ch01 소프트웨어 설계]

>01 요구사항 확인: 현행 시스템 분석

Q1 정답 : 3

현행 시스템 파악 과정에 업무 프로세스 파악은 포함되지 않습니다.

[Hint]

현행 시스템 파악 절차

1. 시스템 구성 /기능/인터페이스 파악
2. 아키텍처 및 소프트웨어 구성 파악
3. 하드웨어 및 네트워크 구성 파악

Q2 정답 : 3

공급 벤더들의 안정적인 기술 지원은 “기술 지원”에 관한 내용입니다.

[Hint]

DBMS 고려사항

- 가용성 - 성능
- 기술 지원 - 상호 호환성 - 구축 비용

>01.요구사항 확인 : 요구사항 확인

Q3 정답 : 2

개발 인력의 경험이 적고 고객 및 사용자 지식의 정도도 낮을 경우는 브레인스토밍, 워크샵, 프로토타이핑 기법을 선택합니다.

[Hint]

요구사항 도출기법

- 고객의 발표 - 문헌조사
- 업무절차 및 양식조사
- 설문지 - 인터뷰
- JAD(joint Application Development)
- 프로토타이핑 - 사용자 관찰
- 롤플레이팅 등

* 프로토타이핑

프로토타입은 사용자 요구사항을 기반으로 실제로 동작하는 것처럼 만든 동적인 형태의 모형이며, 요구사항에 대한 확인수단이면서 프로토타이핑을 수행하면서 새로운 요구사항이 도출될 수 있음

Q4 정답 : 3

정형분석에 관한 설명입니다.

[Hint]

요구사항 분석기법

- 요구사항 분류
- 개념 모델링 - 요구사항 할당
- 요구사항 협상 - 정형 분석

* 정형 분석(Formal Analysis)

구문과 의미를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 분석하는 과정

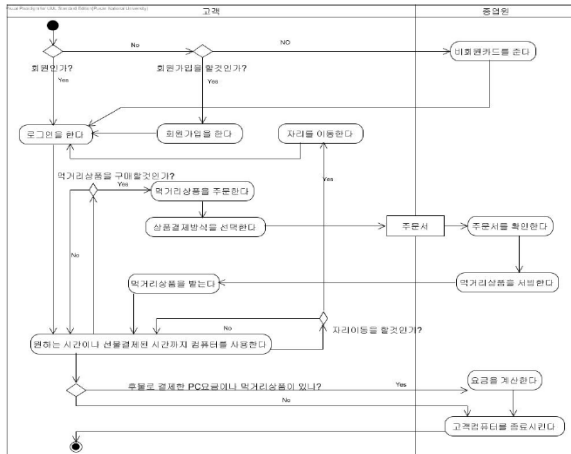
Q5 정답 : 4

요구사항 확인은 요구사항이 자원에 할당되기 전에 문제 파악을 위한 검증을 수행합니다.

>01.요구사항 확인: 분석모델확인

Q6 정답 : 1

시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현한 것은 Use Case Diagram 입니다.



>02.화면 설계 : UI 요구사항 확인

Q7 정답 : 2

사용자의 요구사항에 맞게 화면 구조나 화면 배치 등을 설계하는 것은 **UI 설계** 입니다.

Q8 정답 : 3

회복성(**Recoverability**)은 신뢰성(**Reliability**) 품질 특성에 속하는 상세 품질 요구에 해당됩니다.

[Hint]

소프트웨어 품질 특성과 평가를 위한 국제 표준 지침
ISO/IEC 9126
소프트웨어 품질특성 **6** 가지

- 기능성 - 신뢰성
- 사용성 - 효율성
- 유지 보수성 - 이식성

Q9 정답 : 1

누구나 쉽게 이해하고 사용 가능한 것은 **UI** 의 설계 원칙 **4** 가지 중 ‘직관성’입니다.

[Hint]

UI 의 설계원칙

- 직관성 :
누구나 쉽게 이해하고 사용 가능
- 유효성 :

사용자의 목적을 정확하게 달성

- 학습성 :
누구나 쉽게 배우고 익힘
- 유연성 :
사용자의 요구사항을 최대한 수용하며, 오류를 최소화

>02.화면 설계 : UI 설계

Q10 정답 : 1

UI 시나리오는 시스템의 기능 보다 사용자의 태스크에 초점을 맞춰 기술합니다.

[Hint]

UI 시나리오

사용자가 경험하게 될 이야기를 미리 그려봄으로써 서비스 전체의 윤곽을 표현한 것이며, 웹(앱)을 사용하는 모든 단계마다 일어나는 사용자와 서비스의 상호 작용을 기술

Q11 정답 : 4

사용자가 시스템이나 서비스를 이용하면서 느끼고 생각하게 되는 총체적인 경험은 **UX(User Experience)**입니다.

Q12 정답 : 4

공통 모듈은 외부 모듈 간의 결합도는 낮고 응집도는 높아야 한다.

Q13 정답 : 3

이상적인 설계 품질의 소프트웨어는 응집도는 높고 결합도는 낮습니다.

[Hint]

응집도(**Cohesion**)

명령어나 호출문 등 모듈의 내부 요소들 간의 서로 관련된 정도

결합도(**Coupling**)

모듈 간의 상호 의존하는 정도 또는 두 모듈 사이의 연관 정도

>03. 애플리케이션 설계 : 공통 모듈 설계

Q14 정답 : 1

기능은 순차적으로 통신해야, 절차적 시간이 논리적이 되고 우연이 낮음.

>03. 애플리케이션 설계 : 객체지향 설계

Q15 정답 : 2

상속은 상위 클래스로부터 메소드와 속성을 물려받는 것입니다.

[Hint]

객체지향 특징(Cohesion)

- 추상화 - 상속
- 다형성 - 캡슐화
- 정보은닉

Q16 정답 : 3

집합관계는 컴포넌트 클래스와 전체 클래스를 선으로 잇고, 빈 마름모꼴을 전체 클래스 쪽에 붙여서 나타냅니다.

Q17 정답 : 2

Factory Method 패턴에 관한 설명입니다.

“목적과 범위에 따른 분류”

- 생성패턴: 객체의 생성과정에 관여하는 패턴
- 구조패턴: 클래스나 객체의 합성에 관한 패턴
- 행위패턴: 클래스나 객체들의 상호작용과 책임을 분산하는 방법

목적 범위	생성	구조	행위
클래스	Factory Method	Adapter	<ul style="list-style-type: none"> • Interpreter • Template Method
객체	<ul style="list-style-type: none"> • Abstraction Factory • Builder • Prototype • Singleton 	<ul style="list-style-type: none"> • Adapter • Bridge • Composite • Decorator • Façade • Flyweight • Proxy 	<ul style="list-style-type: none"> • Chain of Responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

>04. 인터페이스 설계 : 인터페이스 요구사항 확인

Q18 정답 : 2

기능적 요구사항은 시스템의 입력과 출력 및 그들 사이의 처리 과정과 관련된

기능 수행에 해당되며, 정보 등록은 입력과 관련된 기능적 요구사항입니다.

[Hint]

비기능적 요구사항 - 성능(응답 시간, \처리량)
 - 사용의 용이성 - 신뢰도
 - 보안성 - 운용상의 제약,
 - 안전성 등
 행위적 특성이나 시스템 전반과 관련된 요구 사항

>04. 인터페이스 설계 : 인터페이스 대상 식별

Q19 정답 : 4

인터페이스 설계 데이터베이스 산출물은 개체 정의서, 테이블 정의서, 코드 정의서입니다.

> 04. 인터페이스 설계 : 인터페이스 상세 설계

Q20 정답 : 3

배치(Batch) 방식은 일정 시간 또는 일정량을 모았다가 한꺼번에 처리하며 대량의 데이터 처리에 적합합니다.

[Ch02 소프트웨어 개발]

>02.데이터 입출력 구현 : 논리 데이터저장소 확인

Q1 정답 : 1

입력 제한 테크는 Scroll이고 출력 제한 테크는 Shelf입니다.

Q2 정답 : 1

스택에서 **push**는 자료를 넣고 **pop**은 자료를 빼는 동작입니다.따라서, 정답은 **B D C A** 입니다.

Q3 정답 : 2

중위순회는 원.중.오 순이며, **D-B-A-E-C-F** 순입니다.

Q4 정답 : 3

데이터베이스 설계 단계는 요구사항 정의-개념 모델링-논리 모델링-물리 모델링
- 데이터베이스 구축 순입니다.

[Hint]

데이터베이스 설계단계

- ① 요구사항 정의 ② 개념 모델링
- ③ 논리 모델링 ④ 물리 모델링
- ⑤ 데이터베이스 구축

Q5 정답 : 1

데이터의 중복으로 인하여 릴레이션에 대한 관계연산을 처리할 때 발생할 수 있는 부작용은 삽입-수정-삭제이상(anomaly)입니다.

[Hint]

이상(anomaly) 현상

불필요한 데이터 중복으로 인해 릴레이션에 대한 데이터 삽입 수정 삭제 연산을 수행할 때 발생할 수 있는 부작용

Q6 정답 : 2

정규화는 데이터의 중복으로 인하여 릴레이션에 대한 관계연산을 처리할 때 발생할 수 있는 삽입 수정 삭제 이상의 발생을 제거한다.

>02.데이터 입출력 구현 : 물리 데이터저장소 설계

Q7 정답 : 2

내부 스키마는 전체 데이터베이스의 물리적 저장 형태를 기술합니다.

[Hint]

데이터베이스 스키마(Schema)

데이터베이스에서 자료의 구조, 자료의 표현 방법, 자료 간의 관계를 형식 언어로 정의한 구조

스키마(Schema) 종류

- 외부 스키마

프로그래머나 사용자의 입장에서 데이터베이스의 모습으로 조직의 일부분을 정의한 것

- 개념 스키마

모든 응용 시스템과 사용자들이필요로 하는 데이터를 통합한 조직 전체의 데이터베이스 구조를 논리적으로 정의한 것

- 내부 스키마

전체 데이터베이스의 물리적 저장 형태를 기술

Q8 정답 : 3

레코드의 삽입과 삭제가 수시로 일어나는 경우에는 인덱스의 개수를 최소로 하는 것이 효율적입니다.

[Hint]

인덱스(Index)

데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성

되는 데이터 구조

> 01. 데이터 입출력 구현 : 데이터 조작
프로시저 작성

Q9 정답 : 2

한 가지 동작을 수행하는 작은 기능을 단위
기능이라고 하며 단위 기능을 모듈은 단위
모듈입니다.

> 02. 통합 구현 : 모듈 구현

Q10 정답 : 1

단위 모듈에 초점을 맞춰 테스트하는 단계는
단위 테스트입니다.

[Hint]

소프트웨어 개발 단계 테스트

- ① 단위 테스트 ② 통합 테스트
- ③ 시스템 테스트 ④ 인수 테스트

> 02. 통합 구현 : 모듈 구현 관리

Q11 정답 : 2

형상관리는 소프트웨어 개발과정에서
소프트웨어의 변경사항을 관리하는
것입니다.

> 03. 제품소프트웨어 패키징 : 제품소프트웨어
패키징

Q12 정답 : 1

릴리즈 노트(Release Note)에 관한
설명입니다.

> 03. 제품소프트웨어 패키징 : 제품소프트웨어
버전 관리

Q13 정답 : 3

안드로이드 개발 환경에서 빌드 자동화 도구로

사용되는 것은 Gradle입니다.

[Hint]

Groovy

자바에 Python, Ruby, Smalltalk 등의
장점을 결합한 동적 객체 지향 프로그래밍
언어

> 04. 애플리케이션 테스트 관리 :
애플리케이션 테스트케이스 설계

Q14 정답 : 3

블랙박스 검사는 프로그램의 소스를 보지 않고
입력과 출력 결과로 테스트하며
동치 분할검사, 경계값 분석, 원인 효과
그래픽 기법, 비교 검사가 해당됩니다.

[Hint]

블랙박스테스트

- 동치 분할 테스트 -경계 값 테스트
- 원인 효과 그래프 테스트
- 오류 예측 테스트 -비교 테스트

화이트박스테스트

- 데이터 흐름 검사 -루프 검사
- 분기 검사 -조건 검사

Q15 정답 : 1

작은 단위에서 큰 단위로 테스트한 후 시스템
전체를 테스트하고 사용자가
인수 하기 전에 마지막으로 인수 테스트를
수행합니다.

Q16 정답 : 3

알파 검사는 가상의 실험환경에서 개발자가
테스트하며, 베타 검사는 일반 사용자들의
사용환경에서 테스트하는 방법입니다.

[Hint]

인수 테스트

알파 검사

개발자가 가상의 실험 환경을 조성하여

자체적으로 사용하며 테스트

베타 검사

베타 테스터를 모집하여 사용자들의 사용 환경에서 테스트

>04. 애플리케이션 테스트 관리 :

애플리케이션 성능 개선

Q17 정답 : 2

삽입 정렬은 왼쪽부터 순차적으로 모든 인접한 수를 비교한 후 적절한 위치에 삽입해 가며 정렬하는 방식입니다.

[Hint]

정렬 알고리즘

- 삽입 정렬

왼쪽부터 순차적으로 모든 인접한 수를 비교한 후 적절한 위치에 삽입하며 정렬하는 방식

- 버블 정렬

왼쪽부터 순차적으로 두 수를 서로 비교해 가며 큰 수를 가장 뒤로 보내며 정렬하는 방식

- 선택 정렬

왼쪽부터 순차적으로 가장 작은 수를 찾아 해당 자리수만 교환하며 정렬

Q18 정답 : 2

이진 검색 방법은 정렬된 데이터 집합을 이분화 하면서 탐색하는 방법입니다.

>05. 인터페이스 구현 : 인터페이스 설계

확인

Q19 정답 : 3

Message Bus는 애플리케이션 사이에 미들웨어(Bus)를 두어 처리하는 방식입니다.

>05. 인터페이스 구현 : 인터페이스 기능 구현

Q20 정답 : 4

인터페이스 보안 기능 적용 대상은 Application, Database, Network입니다.

[Ch03 데이터베이스 구축]

>01. SQL 응용 : 절차형 SQL 작성

Q1 정답 : 2

절차형 **SQL**의 종류 중 사용자 정의 함수는 사용자가 직접 함수를 정의하여 실행시킬 수 있습니다.

>01. SQL 응용 : 응용 SQL 작성

Q2 정답 : 1

보기의 **BETWEEN**연산은 90이상 95이하에 해당되는 점수를 뽑습니다.

>02. SQL 활용 : 기본 SQL 작성

Q3 정답 : 1

DML에 포함되는 것은 **SELECT, UPDATE, INSERT, DELETE** 입니다.

[Hint]

DML

- **SELECT - INSERT**
- **UPDATE - DELETE**

DDL

- **CREATE - ALTER**
- **DROP - RENAME**
- **TRUNCATE**

DCL

- **GRANT - REVOKE**
- **COMMIT - ROLLBACK**

Q4 정답 : 4

DDL(Data Definition Language)는 **CREATE, ALTER, DROP** 등입니다.

Q5 정답 : 1

트랜잭션 연산 중 트랜잭션을 실행되기 전 상태로 되돌리는 것은 **ROLLBACK**입니다.

[Hint]

트랜잭션의 연산

Commit

하나의 트랜잭션이 끝났다는 것을 알려주기 위해 사용하는 연산

Rollback

하나의 트랜잭션 처리가 비정상적으로 종료되어 트랜잭션의 원자성이 깨진 경우 트랜잭션을 처음부터 다시 시작하거나 부분적으로만 연산된

결과를 취소시킴

> 02. SQL 활용 : 고급 SQL 작성

Q6 정답 : 4

뷰는 저장장치 내에 논리적으로 존재하며, 뷰가 정의된 기본 테이블이 삭제 되면 뷰는 자동적으로 삭제됩니다.

[Hint]

뷰(View)의 장점

- 논리적 독립성 제공
- 사용자 데이터 관리 용이
- 데이터 보안 용이

뷰(View)의 단점

- 뷰 자체 인덱스 불가
- 뷰 정의 변경 불가
- 데이터 변경 제약 존재

> 03. 논리 데이터베이스 설계 : 관계데이터베이스 모델

Q7 정답 : 2

Degree는 릴레이션의 속성(열)의 개수이고, **Cardinality**는 릴레이션의 튜플(행)의 개수입니다.

[Hint]

차수(**Degree**)

릴레이션에서 속성의 전체 개수

카디널리티(**Cardinality**)

릴레이션에서 튜플의 전체 개수

Q8 정답 : 4

관계 대수 연산자 중 교차곱을 구하는 연산은 카티션 프로덕트입니다.

[Hint]

관계대수 연산자

일반 집합 연산자

- 합집합 - 교집합
- 차집합 - 카티션 프로덕트 (교차곱)

Q9 정답 : 1

시스템 카탈로그는 일반 사용자가 **SQL**을 이용하여

내용을 검색하는 것은가능하나 사용자가
SQL문으로 갱신하는 것은 불가능합니다.

>03. 논리 데이터베이스 설계 : 데이터모델링 및 설계

Q10 정답 : 1

오너-멤버관계로 구성되는 데이터 모델은 네트워크
데이터모델이고
부모-자식관계로 구성되는 데이터 모델은 계층형
데이터모델입니다.

Q11 정답 : 2

정규화는 중복 데이터를 최소화하여 데이터
중복으로 발생하는 이상현상을 제거합니다.

[Hint]

정규화(Normalization)

중복성을 최소화하고 정보의 일관성을 보장하기
위한 개념이며, 릴레이션을 관련 있는
속성들로만 구성되도록 릴레이션(테이블)을
분해하는 과정

Q12 정답 : 4

1정규형에서 2정규형이 되기 위한 제2정규화는
부분 함수 종속성 제거입니다.

[Hint]

정규화(Normalization)

-제1정규화

: 중복 속성 제거

- 제2정규화

: 부분 종속성 제거

- 제3정규화

: 이행 종속성 제거

- 제4정규화

: 다치 종속성 제거

- 제5정규화

: 조인 종속성 제거

>04. 물리 데이터베이스 설계 : 물리요소 조사 분석

Q13 정답 : 4

목표 DBMS에 맞는 스키마 설계는 논리적 설계
단계에서 수행합니다.

[Hint]

데이터베이스 설계 단계

- ① 개념적 설계
- ② 논리적 설계
- ③ 물리적 설계

물리적 설계 시 설계 사항

- 저장 레코드의 양식설계
- 레코드 집중의 분석 및 설계
- 접근경로설계

Q14 정답 : 1

스토리지의 종류는 DAS, NAS, SAN이며 SSD(Solid
State Drive)는 하드디스크를 대체하는 반도체를
이용한 고속의 보조기억장치입니다.

Q15 정답 : 1

위치 투명성은 어느 위치에 분산되어 있든지
관계없이 사용할 수 있는 것이 특징입니다.

[Hint]

분산 데이터베이스4가지 투명성

- 위치투명성
- 중복투명성
- 병행투명성
- 장애투명성

>04. 물리 데이터베이스 설계 : 데이터베이스 물리 속성 설계

Q16 정답 : 3

파티션은 논리적 단위이지만 테이블을
파티셔닝하면 물리적으로 별도의 공간에 데이터가
저장됩니다.

>04. 물리 데이터베이스 설계 : 물리 데이터베이스 모델링

Q17 정답 : 3

도메인 무결성은 튜플을 구성하는 속성의 값이 그 속성이 정의된 도메인에 속한 값이어야 한다는 것을 말합니다.

[Hint]

무결성이란?

데이터의 정확성, 일관성, 유효성, 신뢰성 등과 무효 갂신으로부터 데이터 보호

무결성의 종류

- 개체 무결성
- 도메인 무결성
- 참조 무결성
- 사용자 정의 무결성

Q18 정답 : 2

데이터베이스 레코드를 유일하게 식별할 수 있는 주된 키는 기본키입니다.

[Hint]

키의 개념

데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 튜플들을 서로 구분할 수 있는 기준이 되는 애트리뷰트

키의 종류

- 슈퍼키 - 후보키
- 기본키 - 대체키 등

Q19 정답 : 3

처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우 반정규화 시 중복 테이블을 추가합니다.

[Hint]

반정규화 중복 테이블을 추가하는 경우

- 정규화로 인해 수행 속도가 느려지는 경우
- 많은 범위 데이터를 자주 처리해야 하는 경우
- 특정 범위 데이터만 자주 처리해야 하는 경우
- 처리 범위를 줄이지 않고는 수행 속도를 개선할수 없는 경우

>05. 데이터전환 : 데이터 전환 기술

Q20 정답 : 1

기존 정보시스템에서 데이터를 추출하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환한 후 적재하는 일련의 과정은 데이터 전환입니다.

[Hint]

데이터 전환

운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출(**Extraction**)하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환(**Transformation**)한 후, 적재(**Loading**)하는 일련의 과정

[CH 04 프로그래밍 언어 활용]

>01. 서버 프로그램 구현 : 개발 환경 구축

Q1 정답 : 2

개발 환경 구축은 응용 소프트웨어 개발을 위해 소프트웨어와 하드웨어 장비를 구축하는 것입니다.

Q2 정답 : 4

개발언어의 선정 기준 중 친밀성은 프로그래머가 그 언어를 이해하고 사용할 수 있어야 합니다. 최신 언어를 선택하는 것이 꼭 좋지는 않습니다.

[Hint]

개발언어 선정기준

- 적정성 - 효율성
- 이식성 - 친밀성
- 범용성

>01. 서버프로그램 구현 : 공통 모듈 구현

Q3 정답 : 3

소프트웨어 재사용은 프로젝트 실패의 위험성을 줄일 수 있습니다.

Q4 정답 : 2

결합도는 낮은 것이 좋습니다.

높은 내용은 공통적인 것을 외부로부터 제어한 후 도장 찍은 자료이다.

>01. 서버프로그램 구현 : 서버 프로그램 구현

Q5 정답 : 4

4번은 “입력 데이터 검증 및 표현”에 관한 설명입니다.

[Hint]

소프트웨어 개발 보안 점검 항목

- 입력 데이터 검증 및 표현
- 보안 기능 - 시간 및 상태
- 에러 처리 - 코드 오류
- 캡슐화 - API 오용

>01. 서버 프로그램 구현 : 배치 프로그램 구현

Q6 정답 : 3

사용자와 상호작용 없이 일련의 작업들을 작업단위로 묶어 미리 정해진

일련의 순서에 따라 일괄적으로 처리하는 것은

Batch Program입니다.

[Hint]

API

응용 프로그램 개발 시 운영체제나 프로그래밍 언어 등에 있는 라이브러리를 이용할 수 있도록 규칙 등을 정해 놓은 인터페이스

배치 프로그램

사용자와의 상호 작용 없이 일련의 작업들을 작업

단위로 묶어 정기적으로 반복 수행하거나 정해진 규칙에 따라 일괄 처리하는 것

>02. 프로그래밍 언어 활용 : 기본문법 활용

Q7 정답 : 4

byte는 1Byte, int는 4Byte, short는 2byte, long은 8Byte입니다.

종류	데이터 타입	크기		기억 범위
문자	char	2Byte	16bit	0~65,535
정수	byte	1Byte	8bit	-128~127
	short	2Byte	16bit	-32,768~32,767
	int	4Byte	32bit	-2,147,483,648~2,147,483,648
	long	8Byte	64bit	-9,223,137,203,685,475,808~9,223,137,203,685,475,807
실수	float	4Byte	32bit	1.4X10 ⁻⁴⁵ ~3.4X10 ³⁸
	double	8Byte	64bit	4.9X10 ⁻³²⁴ ~1.8X10 ³⁰⁸
논리	boolean	1Byte	8bit	True 혹은 False

Q8 정답 : 1

대입 → 조건 → 논리 → 관계 → 시프트 →

덧셈, 뺄셈 → 곱셈, 나눗셈 → 부정, 증감 → 괄호

연산자	연산 내용	우선 순위
(), []	괄호, 대괄호	1
!, ~, ++, --	부정, 증감 산술 연산자	2
*, /, %	곱셈, 나눗셈 산술 연산자	3
+, -	덧셈, 뺄셈 산술 연산자	4
<<, >>, >>>	시프트 연산자	5
<, <=, >, >=, ==, !=	관계 연산자	6
&, ^, , &&,	논리 연산자	7
?:	조건 연산자 예 : b*=a>b?a:b; (a=1, b=2)	8
=, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, ~=	대입 연산자	9

Q9 정답 : 2

1단계: $(2*4) = 8$, 2단계: $8+4 = 12$, 3단계: $12/2 = 6$,
4단계: $6+3 = 9$

> 02. 프로그래밍 언어 활용 : 언어특성 활용

Q10 정답 : 2

C언어는 인간 친화적인 고급 언어에 해당됩니다.

Q11 정답 : 2

정보은닉은 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것이며, 2번의 설명은 다형성에 관한 설명입니다.

[Hint]

객체지향 프로그래밍 언어 특징

1. 추상화
2. 상속성
3. 캡슐화
4. 정보은닉
5. 다형성

> 02. 프로그래밍 언어 활용 : 라이브러리 활용

Q12 정답 : 2

제어문자 중 커서를 현재 줄의 처음으로 이동하는 것은 “\r”이며, carriage return을 의미합니다.

제어문자	의미	기능
\n	new line	커서를 다음 줄 앞으로 이동
\b	backspace	커서를 왼쪽으로 한 칸 이동
\t	tab	커서를 일정 간격 띄움
\r	carriage return	커서를 현재 줄의 처음으로 이동
\0	null	널 문자를 출력
\'	single quote	작은 따옴표를 출력
\"	double quote	큰 따옴표를 출력
\\	backslash	역슬러시를 출력
\f	form feed	한 페이지를 넘김

> 03. 응용 SW 기초 기술 활용 : 운영체제 기초 활용

Q13 정답 : 3

원시프로그램을 목적 프로그램으로 변환하는 것은 언어 번역 프로그램입니다.

Q14 정답 : 1

Best Fit을 사용하면 빈 기억 공간의 크기가 20K에 적재하며, 내부 단편화의 크기는 $20-17=3K$ 입니다.

[Hint]

기억장치 배치전략

- 최초 적합(First Fit)
- 최선 적합(Best Fit)
- 최악 적합(Worst Fit)

내부 단편화

사용할 프로그램이 18KB일 때 페이지 크기 4KB씩 나누면 2KB 단편화 발생

Q15 정답 : 4

PCB에 저장되는 정보로는 주기억장치 관리 정보와 입출력 상대정보는 포함되지만 할당되지 않은 주변 장치의 상태 정보는 포함되지 않습니다.

[Hint]

PCB(Process Control Block, 프로세스 제어 블록)

운영체제가 프로세스에 대한 중요한 정보를 저장해 놓는 곳

PCB에 저장되는 정보의 종류

- ① 프로세스의 현재상태(준비, 대기, 실행 등)
- ② 포인터(부모와 자식 프로세스의 주소 기억 등)
- ③ 프로세스 고유 식별자
- ④ 스케줄링 및 프로세스의 우선순위
- ⑤ CPU 레지스터 정보
- ⑥ 주기억장치 관리 정보
- ⑦ 입출력 상대정보
- ⑧ 계정 정보

Q16 정답 : 2

Windows의 MD는 디렉토리 생성 명령어이며

UNIX의 **mkdir** 에 해당되며,
UNIX의 **chmod**는 파일 사용 권한 지정
명령어입니다.

>03. 응용 SW 기초 기술 활용 : 네트워크 기초 활용

Q17 정답 : 3

감쇠된 전송 신호를 증폭시키는 네트워크 장비는
리피터(**Repeater**)입니다.

[Hint]

네트워크 장비

- 라우터 - 게이트웨이
- 리피터 - 브리지
- 허브 - 스위치

Q18 정답 : 3

네트워크 계층의 데이터 전송 단위는
패킷(**Packet**)입니다.

[Hint]

OSI 7 Layer

계층별 데이터단위

7계층 응용 계층

6계층 표현 계층

5계층 세션 계층

: 메시지(**Message**)

4계층 전송 계층

: 세그먼트(**Segment**)

3계층 네트워크

: 패킷(**Packet**)

2계층 데이터링크

: 프레임(**Frame**)

1계층 물리계층

: 비트(**bit**)

Q19 정답 : 3

RIP는 최대 홉 카운트를 **16**홉 이하로 한정합니다.

[Hint]

라우팅 프로토콜의 종류

RIP : 거리 벡터 알고리즘 활용, **16**홉,
소규모 네트워크

OSPF : 링크 상태 알고리즘 사용,
대규모 네트워크

IGRP : 네트워크 상태를 고려하여 라우팅

BGP : 규모가 큰 네트워크의 상호 연결

>03. 응용 SW 기초 기술 활용 : 기본 개발환경 구축

Q20 정답 : 3

현존 시스템과의 연동과 연계를 고려해야 합니다.

[Hint]

패키지(**Package**) 개발

여러 성공사례의 노하우를 기반으로 개발된
제품을 이용하여 시스템을 구축하는 방식

[Ch 05 정보시스템 구축관리]

>01. 소프트웨어개발 방법론 활용 :

소프트웨어개발 방법론 선정

Q1 정답 : 3

폭포수 모형과 같이 선형 순차 모형의 개발 순서는
분석 → 설계 → 구현 → 테스트 → 유지보수
순입니다.

[Hint]

소프트웨어 생명 주기 모형

- 폭포수 모형 - 프로토타입 모형
- 나선형 모형 - 애자일 모형

선형 순차 모형 개발 순서

- ① 분석
- ② 설계
- ③ 구현
- ④ 테스트
- ⑤ 유지보수

Q2 정답 : 1

프로토타입 모형은 최종 프로토타입을 버리고 새로
소프트웨어를 개발하는 실험적 프로토타입 모형과
프로토타입을 지속적으로 발전시켜 개발하는
진화적 프로토타입 모형으로 나뉘지며, 프로토타입
개발에 비유적인 언어나 알고리즘을 사용하지
않습니다.

Q3 정답 : 3

COCOMO 개발 유형별로 노력(**MM**)과 개발
기간(**TDEV**)를 계산하는 공식이 상이합니다.

[Hint]

COCOMO 모델 개발유형

Organic Mode

(조직형)

중.소 규모 5만 라인 이하

Semi-detached

(반분리형)

Organic과 **Embedded** 중간형 30만 라인 이하

Embedded

(내장형)

초대형 규모의 트랜잭션 처리 시스템이나
운영체제 등 30만 라인 이상

* **COCOMO** 개발

유형별 노력과 개발기간 산정공식이 상이함

Q4 정답 : 2

Putnam 모형을 기초로 만든 비용 산정 자동화
추정 도구는 **SLIM**이며, **FP** 모형을 기초로 만든
비용 산정 자동화 추정 도구는 **ESTIMACS**입니다.

[Hint]

비용 산정 자동화 추정 도구

① **SLIM**

Rayleigh-Norden 곡선과 **Putnam** 예측 모델을
기초로 개발

② **ESTIMACS**

FP 모형을 기초로 개발

Q5 정답 : 1

소프트웨어 프로젝트의 비용과 노력 측정을
위해서는 프로젝트 요소, 자원요소, 생산성 요소를
고려한 적합한 비용 산정 모델과 자동화
측정도구를 이용합니다.

[Hint]

소프트웨어 비용 결정 요소

프로젝트 요소

제품 복잡도, 시스템 크기, 요구되는 신뢰도

자원 요소

인적 자원, 하드웨어 자원, 소프트웨어 자원

생산성 요소

개발자 능력, 개발 기간

>01. 소프트웨어개발 방법론 활용 : 소프트웨어개발 방법론 테일러링

Q6 정답 : 4

소프트웨어 개발 방법론 테일러링 기법은 프로젝트
규모와 복잡도, 프로젝트 구성원, 팀내 방법론 지원,
자동화에 따른 테일러링 기법이 있습니다.

[Hint]

테일러링(Tailoring)

사전적 의미는 ‘재단, 양복업’ 으로 표준을
기반으로 실제 업무에서 여건에 맞게 수정 보완
하는 것

소프트웨어 개발 방법론 테일러링 기법

- ①프로젝트 규모와 복잡도에 따른 테일러링
- ②프로젝트 구성원에 따른 테일러링
- ③팀내 방법론 지원에 따른 테일러링
- ④자동화에 따른 테일러링

>02. IT프로젝트 정보시스템 구축 관리 : 네트워크 구축 관리

Q7 정답 : 4

휴대용 기기로 언제나 자유롭게 네트워크에 접속하여 업무를 처리할 수 있는 환경은 모바일 컴퓨팅(Mobile Computing)입니다.

[Hint]

스마트 그리드

전기의 생산부터 소비까지의 전 과정에 정보통신기술을 접목하여 에너지 효율을 높이는 지능형 전력망 시스템

Q8 정답 : 3

LAN의 네트워크 형태는 주로 버스형과 링형을 사용합니다.

[Hint]

규모에 따른 네트워크 분류

1. LAN
2. MAN
3. WAN

네트워크 구성 형태(Topology)

1. 스타형
2. 링형
3. 버스형
4. 트리형
5. 망형

>02. IT프로젝트 정보시스템 구축 관리 : SW 구축 관리

Q9 정답 : 2

아키텍트(Architect)의 소프트웨어 개발 보안 활동에 관한 설명입니다.

>02. IT프로젝트 정보시스템 구축 관리 : HW 구축 관리

Q10 정답 : 4

한번의 기록만으로 자료를 영구 보관할 수 있는 광 저장 장치는 M-DISC입니다.

[Hint]

멤스(MEMS; Micro-Electro Mechanical Systems)

초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치

>02. IT프로젝트 정보시스템 구축 관리 : DB 구축 관리

Q11 정답 : 3

DB 관련 신기술 중 빅 데이터(Big Data)에 관한 설명입니다.

[Hint]

메타 데이터

일련의 데이터를 정의하고 설명해 주는 데이터

브로드 데이터

대량의 자료를 뜻하는 빅데이터와 달리 다양한 정보를 뜻하는 것

Q12 정답 : 1

병행제어 목적은 사용자에 대한 응답 시간 최소화입니다.

[Hint]

병행제어

동시에 여러 개의 트랜잭션을 병행 수행할 때, 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호작용을 제어

병행제어의 목적

- ①데이터베이스의 공유를 최대화
- ②시스템의 활용도 최대화
- ③데이터베이스의 일관성 유지
- ④사용자에 대한 응답 시간을 최소화

>03. 소프트웨어 개발 보안 구축 : SW개발 보안 설계

Q13 정답 : 3

Secure SDLC는 소프트웨어 유지보수 단계에서 보안 이슈를 해결하기 위해 소모되는 비용을 최소화하기 위해 등장했습니다.

Q14 정답 : 2

주요 보안 요소 중 오직 인가된 사용자만 수정할 수 있는 것은 무결성(Integrity) 입니다.

[Hint]

주요 보안요소(Security Elements)

- 기밀성

인가된 사용자만 접근 허용

- 무결성

인가된 사용자만 수정 가능

- 가용성

인가받은 사용자는 언제라도 사용가능
그 외에 인증, 부인방지가 포함됨

Q15 정답 : 3

인증과정 없이 중요한 정보 또는 기능에 접근하는 것을 방지하기 위해서는 중요한 정보나 기능을 수행하는 페이지에서 재인증 기능을 수행하도록 합니다.

[Hint]

보안기능의 보안 약점 종류

- 적절한 인증 없이 중요기능 허용
- 중요한 자원에 대한 잘못된 권한 설정
- 하드코딩된 비밀번호
- 중요정보 평문 저장 및 전송
- 취약한 암호화 알고리즘 사용
- 부적절한 인가

Q16 정답 : 2

부적절한 자원 해제는 코드 오류와 관련된 보안 약점입니다.

[Hint]

에러 처리

소프트웨어 실행 중 발생할 수 있는 오류들을

사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안

점검 항목들

- 오류 메시지를 통한 정보노출
- 부적절한 예외 처리
- 오류 상황 대응 부재

>03. 소프트웨어 개발 보안 구축 : SW개발 보안 구현

Q17 정답 : 1

RSA(Rivest Shamir Adleman)암호화 알고리즘은 공개키 암호화 알고리즘이며, 나머지는 개인키 암호화 알고리즘입니다.

[Hint]

비밀키(개인키)

암호화 알고리즘 종류

- DES
- SEED
- AES
- ARIA

공개키 암호화

알고리즘의 종류

- RSA
- DSA

>04. 시스템 보안 구축 : 시스템 보안 설계

Q18 정답 : 1

여러 대의 공격자를 분산적으로 배치해 동시에 서비스 거부 공격을 하는 방법은 분산 서비스 거부 공격(Distributed DoS) 또는 디도스(DDoS)입니다.

Q19 정답 : 1

3-way-hanshake와 관련된 서비스 공격 유형은 **SYN Flooding**입니다.

[Hint]

*3-way-handshake

데이터를 전송하기 전에 먼저 정확한 전송을 보장하기 위해 상대방 컴퓨터와 사전에 세션을 수립하는 과정

1단계 :

Client > Server SYN(synchronize sequence numbers) 패킷 전송

2단계 :

Server > Client SYN+ACK 패킷 전송

3단계 :

Client > Server

ACK(acknowledgment) 패킷 전송

Q20 정답 : 2

방화벽(**Firewall**)에 관한 설명입니다.