

출제기준(필기)

직무 분야	정보통신	중직무 분야	정보기술	자격 종목	정보처리산업기사	적용 기간	2022.1.1.~2024.12.31.
○ 직무내용 : 정보시스템 기반 기술을 활용하여 소프트웨어 기능에 대한 구현 및 테스트를 수행하고 사용자 에게 배포하는 직무이다.							
필기검정방법	객관식		문제수	60		시험시간	1시간 30분
필기과목명	문제수	주요항목	세부항목		세세항목		
정보시스템 기반 기술	20	1. 응용 SW 기초 기술 활용	1. 운영체제 기초 활용		1. 운영체제 개념 2. 운영체제 종류 및 특징 3. 운영체제 기본명령어 4. 셸 스크립트(Shell Script)		
			2. 네트워크 기초 활용		1. 프로토콜 2. OSI-7 계층 3. 인터넷 구성 4. TCP/UDP 5. IP 주소 체계		
			3. 기본 개발환경 구축		1. 개발자 개발환경 구축 2. 개발도구 설치 3. 라이브러리 패키지 설치 4. 소프트웨어 버전관리 도구 설치		
		2. 애플리케이션 설계	1. 공통 모듈 설계		1. 설계 모델링 개념 2. 소프트웨어 아키텍처 개념		
			2. 객체지향 설계		1. 객체지향(OOP) 이해 2. 디자인패턴 개념		
		3. 테스트 및 배포	1. 애플리케이션 테스트 수행		1. 통합개발환경(IDE) 도구 활용 2. 프로그램 디버깅 3. 단위테스트 4. 통합테스트 5. 애플리케이션 결함 관리		
			2. UI 테스트		1. UI/UX 개념 2. UI 테스트 기법		
			3. 애플리케이션 배포		1. 소스코드 검증 기법 2. 형상관리		

필기과목명	문제수	주요항목	세부항목	세세항목
프로그래밍 언어 활용	20	1. 프로그래밍 언어 활용	1. 기본문법 활용	3. 애플리케이션 빌드 4. 애플리케이션 배포 도구
			2. 언어특성 활용	1. 데이터 타입 2. 변수 3. 연산자 및 명령문
			3. 라이브러리 활용	1. 절차적 프로그래밍 언어 2. 객체지향 프로그래밍 언어 3. 스크립트 언어
		2. 프로그램 구현	1. 개발환경 구축	1. 데이터 입출력 2. 파일 입출력 3. 예외 처리 4. 프로토타입
			2. 공통 모듈 구현	1. 개발환경 구축 2. 서버 프레임워크 이해
			3. 화면 구현	1. 재사용 2. 모듈화 기법 3. 결함도 4. 응집도
			4. 서버 프로그램 구현	1. UI 프레임워크 2. HTML, CSS, JAVASCRIPT
데이터베이스 활용	20	1. 데이터베이스 이해	1. 데이터베이스 기초 활용	1. 데이터베이스 개념 2. 데이터베이스 유형 및 특징 3. 데이터 무결성 4. 엔티티 및 릴레이션
			2. 논리 데이터베이스 설계	1. 자료구조의 이해 2. E-R 모델링 이해 3. 정규화 이해 4. 데이터 간 관계

필기과목명	문제수	주요항목	세부항목	세세항목
				5. 데이터 간 제약조건
		3. 물리 데이터베이스 설계	1. 물리 데이터저장소	
		2. SQL활용	1. 기본 SQL 작성	1. DDL(Data Definition Language) 2. DML(Data Manipulation Language) 3. DCL(Data Control Language) 4. TCL(Transaction Control Language)
			2. 고급 SQL 작성	1. 집합연산자 2. 조인, 서브쿼리 3. 뷰 4. 인덱스
		3. 데이터베이스 프로그래밍	3. 데이터 조작 프로시저 작성	1. 프로시저 이해 2. 프로시저 오류처리 이해
			4. 데이터 조작 프로시저 최적화	1. 쿼리 성능 측정 도구

출제기준(실기)

직무분야	정보통신	중직무분야	정보기술	자격종목	정보처리산업기사	적용기간	2022.1.1.~2024.12.31.
○ 직무내용 : 정보시스템 기반 기술을 활용하여 소프트웨어의 기능에 관한 구현 및 테스트를 수행하고 사용자에게 배포하는 직무이다. ○ 수행준거 : 1. 응용소프트웨어 개발을 위하여 운영체제, 데이터베이스, 네트워크의 기초 기술을 적용하고 응용개발에 필요한 환경을 구축할 수 있다. 2. UI 표준을 검증하기 위하여 사용성 테스트 계획, 수행, 분석, 결과 보고를 수행할 수 있다. 3. UI 요구사항을 확인하여 설계한 UI 설계를 기반으로 화면을 구현할 수 있다. 4. 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현할 수 있다. 5. 관계형 데이터베이스에서 SQL을 사용하여 목적에 적합한 데이터를 정의하고, 조작하며, 제어할 수 있다. 6. 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트 케이스에 따라 테스트를 수행하고 결함을 조치할 수 있다. 7. 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포할 수 있다.							
실기검정방법		필답형		시험시간		2시간 30분	
실기과목명		주요항목		세부항목		세세항목	
정보처리 실무		1. 응용 SW 기초 기술 활용		1. 운영체제 기초 활용하기 2. 데이터베이스 기초 활용하기 3. 네트워크 기초 활용하기 4. 기본 개발환경 구축하기		1. 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다. 2. CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다. 3. 운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다. 1. 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다. 2. 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다. 3. 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다. 1. 네트워크 계층구조에서 각 층의 역할을 설명할 수 있다. 2. 응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다. 3. 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다. 1. 응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다. 2. 응용개발에 필요한 개발도구를 설치하고 운용할 수 있다. 3. 웹서버, DB서버 등 응용개발에 필요한 기반 서	

실기과목명	주요항목	세부항목	세세항목
	2. UI 테스트	1. 사용성 테스트 계획하기	버를 설치하고 운용할 수 있다. 1. 구현된 UI의 사용성을 검증하기 위하여 적합한 테스트 기법을 선정할 수 있다. 2. 선정된 기법에 적합한 테스트 환경을 구축할 수 있다. 3. 사용성 테스트를 수행하기 위하여 계획서를 작성할 수 있다.
		2. 사용성 테스트 수행하기	1. 선정된 테스트 기법에 맞는 도구를 활용하여 테스트를 수행할 수 있다. 2. 테스트 자료를 수집하여 사용성 평가 분석서를 작성할 수 있다. 3. 사용성 평가 분석서를 토대로 이슈 사항을 도출할 수 있다.
		3. 테스트 결과 보고하기	1. UI 개선 방안을 토대로 수정 계획을 수립할 수 있다. 2. UI 개선 방안과 수립된 수정 계획을 바탕으로 결과 보고서를 작성할 수 있다.
	3. 화면 구현	1. UI 설계 확인하기	1. 설계된 화면과 품의 흐름을 확인하고, 제약사항과 화면의 품 흐름을 구현에 반영하도록 설계를 확인 할 수 있다. 2. UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다. 3. 구현을 위해 하위 시스템 단위의 내·외부 화면과 품을 설계를 확인할 수 있다.
		2. UI 구현하기	1. 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다. 2. 확인된 화면과 품 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 품의 흐름 제어를 구현할 수 있다 3. 확인된 화면과 품 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 품, 메뉴, 흐름을 구현할 수 있다. 4. 구현된 화면, 품, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.
	4. 프로그래밍 언어 활용	1. 기본문법 활용하기	1. 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다. 2. 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다. 3. 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.
		2. 언어특성 활용하기	1. 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.

실기과목명	주요항목	세부항목	세세항목
		3. 라이브러리 활용하기	2. 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다. 3. 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용 할 수 있다. 1. 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다. 2. 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다. 3. 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.
	5. SQL활용	1. 기본 SQL 작성하기	1. 테이블의 구조와 제약조건을 생성, 삭제하고 수정하는 DDL(Data Definition Language) 명령문을 작성할 수 있다. 2. 한 개의 테이블에 대해 데이터를 삽입, 수정, 삭제하고 행을 조회하는 DML(Data Manipulation Language) 명령문을 작성할 수 있다. 3. 업무단위인 트랜잭션의 완료와 취소를 위한 DCL(Data Control Language) 명령문을 작성할 수 있다. 4. 생성된 테이블의 목록, 테이블의 구조와 제약조건을 파악하기 위해 데이터사전을 조회하는 명령문을 작성할 수 있다.
		2. 고급 SQL 작성하기	1. 테이블 조회 시간을 단축하기 사용하는 인덱스의 개념을 이해하고 인덱스를 생성하는 DDL(Data Definition Language)명령문을 작성할 수 있다. 2. 먼저 생성된 테이블들을 이용하여 새로운 테이블과 뷰를 생성하는 DDL(Data Definition Language)명령문을 작성할 수 있다. 3. 조인, 서브쿼리, 집합연산자를 사용하여 두 개 이상의 테이블로부터 데이터를 조회하는 DML(Data Manipulation Language)명령문을 작성할 수 있다.
	6. 애플리케이션 테스트 수행	1. 애플리케이션 테스트 수행하기	1. 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. 2. 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다. 3. 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.
		2. 애플리케이션 결함 조치하기	1. 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다. 2. 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.

실기과목명	주요항목	세부항목	세세항목
	7. 애플리케이션 배포	1. 애플리케이션 배포 환경 구성하기	3. 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다. 1. 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다. 2. 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다. 3. 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다. 4. 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.
		2. 애플리케이션 소스 검증하기	1. 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다. 2. 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다. 3. 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.
		3. 애플리케이션 빌드하기	1. 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다. 2. 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다. 3. 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다. 4. 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지 여부를 확인할 수 있다. 5. 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악할 수 있다.
		4. 애플리케이션 배포하기	1. 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다. 2. 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다. 3. 애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다. 4. 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.