2024 소프트웨어 시스템 설계 및 개발 I

학과:
학번: 이름:

○ 본 과제는 풀이 후 출력하여 수업 시간에 제출 바랍니다.
- 제출 마감 : (월 분반) 5/20 수업, (수 분반) 5/22 수업

○ 제출 시 10점, 미제출시 0점입니다. 제출 지연 시 감점이 있습니다.

○ 본 과제의 문제 중에서 지문만 조금씩 변형하여 중간고사를 제출할 예정입니다.

---

1. 소프트웨어 공학의 목적으로 가장 거리가 먼 내용을 고르시오.
(1) 생산성 향상
(2) 소프트웨어 품질 보장
(3) 프로젝트의 시간과 비용 최적화
(4) 무료 소프트웨어의 제작과 배포

2. 소프트웨어 공학의 다양한 전략 중 가장 거리가 먼 내용을 고르시오.
(1) 요구사항 관리
(2) 설계 및 분석
(3) 레거시 코드 분석
(4) 프로젝트 관리 도구 사용

3. 애자일 개발 방법론에 대한 설명 중 가장 알맞은 내용을 고르시오.
(1) 모든 단계를 순서대로 수행한다고 하여 폭포수 모델이라고도 부른다.
(2) 실행 가능한 단위로 짧은 개발 주기를 가진다.
(3) 팀과의 협업보다 개인 개발에 알맞은 개발 방법론이다.
(4) 고객의 요구사항 변경에 유연하게 대처하기는 힘든 방법론이다.

4. 버전 관리에 대한 설명 중 가장 거리가 먼 내용을 고르시오.
(1) 대표적인 버전관리 소프트웨어로는 git이 있다.
(2) 소프트웨어 개발 과정에서 발생하는 소스코드의 변경사항을 추적할 수 있다.
(3) 팀원 간 협업을 위해서 활용할 수 있다.
(4) 한번 합쳐진 소스코드는 변경이 불가함으로 유의해서 병합해야 한다.

5. (주관식) git의 명령어 중 commit과 push의 차이를 간략하게 설명하시오.

6. 동기와 비동기에 대해 바르게 설명한 것은 무엇인가?

(1) 비동기 방식은 작업의 완료를 기다리지 않기 때문에 여러 작업을 동시에 처리할 수 있으나 작업 완료의 순서가 보장되지 않아 코드의 흐름이 복잡해질 수 있다.

(2) 동기 방식은 다른 작업이 완료될 때까지 다른 작업을 시작하지 않기 때문에 자원을 효율적으로 사용할 수 있다.

(3) 비동기 방식은 동시에 여러작업을 처리할 수 있어 성능이 저하된다.

(4) 동기 코드는 콜백 함수나 이벤트 등을 사용하지 않기 때문에 로직의 흐름이 비동기에 비해 상대적으로 복잡하다.

7. 아래 소스코드의 실행결과는 무엇인가?

```
const text = “hello”;
console.log(text.substring(1, 3));
```

(1) hell

(2) lo

(3) el

(4) NaN

8. 사용자가 입력한 값을 “text”에 저장하고 출력하도록 소스코드를 작성하고 싶다. 아래 소스코드의 빈칸에 알맞은 값은 무엇인가?

```
const readline = require(‘readline’);
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});
_________.question(‘텍스트를 입력해주세요:’, (text) => {
  console.log(text);
});
```

(1) readline

(2) rl

(3) input

(4) output

9. (주관식) 아래의 소스코드를 실행하여 각각 10과 1000000을 입력해보았다. 이 소스코드는 무엇을 하기 위한 소스코드이며, 두 실행 결과의 차이가 무엇인지 간략히 설명하시오.

```
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.question('주사위를 굴릴 횟수를 입력해주세요 (10~1000000): ', (input) => {
  const rolls = parseInt(input);
  const results = rollDice(rolls);
  displayResults(results, rolls);
  rl.close();
});

function rollDice(rolls) {
  const results = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0};
  for (let i = 0; i < rolls; i++) {
    const result = Math.floor(Math.random() * 6) + 1;
    results[result]++;
  }
  return results;
}

function displayResults(results, totalRolls) {
  console.log('주사위 굴리기 결과:');
  for (const [side, count] of Object.entries(results)) {
    const probability = ((count / totalRolls) * 100).toFixed(2);
    console.log(`${side}이(가) 나온 횟수: ${count}, 확률: ${probability}%`);
  }
}
```

10. 아래는 nodejs로 간단한 웹서버를 동작시키는 소스코드이다. 웹 브라우저를 통해 아래의 웹서버에 접근하기 위해 브라우저의 주소창에 입력해야하는 url은 무엇인가?

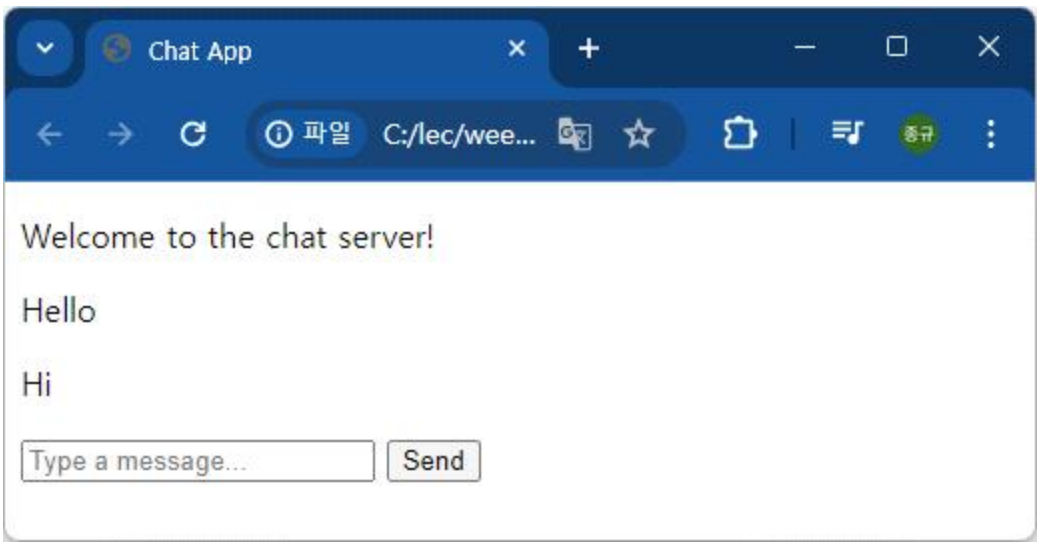
```
const { createServer } = require('http');
const hostname = '127.0.0.1';
const port = 3000;

const server = createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

(1) hostname:3000/
(2) hostname:port
(3) 127.0.0.1:3000
(4) http://${hostname}:${port}/

11. (주관식) 아래의 캡쳐 화면은 간단한 채팅 서비스의 예시 UI이다. 아래 서비스를 개선하기 위해 본인이 생각하는 요구사항을 2가지만 제시해보시오.



12. 아래 HTTP 메서드 중 그 역할이 맞지 않는 것은 무엇인가?
(1) GET : URL 형식으로 서버에 요청
(2) POST : HTTP의 Body에 내용을 담아서 새로운 정보 생성을 서버에 요청
(3) PUT : POST와 비슷하나 주로 기존 정보를 가져오기 위해 요청
(4) DELETE : 정보의 삭제 요청

13. 서버에 요청을 보내어 500 코드를 응답 받았다. 사용자의 대처로 적절한 것은 무엇인가?
(1) 서버의 오류이므로 서비스 관리자에게 연락해서 복구를 요청한다.
(2) 정상적인 응답 코드이므로 서비스를 계속 이용한다.
(3) 사용자의 요청 실수이므로 정상적인 요청을 다시 보낸다.
(4) 다른 페이지로 서비스가 이동하였으므로 이동한 위치를 다시 찾아본다.

14. REST에 대한 설명 중 가장 거리가 먼 것을 고르시오.
(1) 네트워크 아키텍처 스타일 중 하나이다.
(2) REST 아키텍처를 준수한 시스템을 RESTful 하다고 표현한다.
(3) 서버가 클라이언트의 상태를 저장하여 활용하는 것이 원칙이다.
(4) 클라이언트의 각 요청은 필요한 모든 정보를 포함하므로 서버와 독립적이다.

15. 서버가 클라이언트의 상태를 저장하여 발생할 수 있는 문제점이 아닌 것은 무엇인가?
(1) 서버의 복잡성이 증가한다.
(2) 서버의 확장성이 제한된다.
(3) 상태정보 저장을 위해 리소스의 사용이 증가하여 성능에 부담을 준다.
(4) 상태가 저장 돼있으므로 네트워크 오버헤드가 증가한다.

16. REST API에 대한 설명으로 가장 거리가 먼 것을 고르시오.
(1) REST API는 REST 아키텍처를 기반으로 한 API로, HTTP 프로토콜을 사용한다.
(2) GET은 주로 리소스를 조회하는데 사용한다.
(3) nodejs를 통해서 개발하므로 브라우저에 제약적이다.
(4) 쉽고 간편한 개발이 장점으로 확장성과 유연성이 뛰어나다.

17. (주관식) 아래 소스코드를 실행하면 나오는 출력값은 무엇인가?

```
const jsonString = `{
    "address":"Busan",
    "country":"Korea",
    "detail":{
        "address":"Hadan"
    }
}`; // 참고, 문자열은 백틱(` `)으로 감싸져 있음
const jsonObj = JSON.parse(jsonString);
console.log(jsonObj.detail.address);
```

[18~20] 아래는 사용자 관리를 위해 구현한 REST API 코드이다. 각 질문에 답하시오.

```
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

let users = [
{ id: 1, name: 'Kim' },
{ id: 2, name: 'Lee' }];

app.get('/users', (req, res) => {
  res.status(200).json(users);
});

app.get('/users/:id', (req, res) => {
  const user = users.find(u => u.id === parseInt(req.params.id));
  if (user) {
    res.status(200).json(user);
  } else {
    res.status(404).send('User not found');
  }
});

app.post('/users', (req, res) => {
  const user = {
    id: users.length + 1,
    name: req.body.name
  };
  users.push(user);
  res.status(201).json(user);
});

app.delete('/users/:id', (req, res) => {
  users = users.filter(u => u.id !== parseInt(req.params.id));
  res.status(204).send();
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

18. 등록된 모든 사용자 정보를 얻기 위한 GET API의 URL(URI)은 무엇인가?
  (1) localhost:3000/
  (2) localhost:3000/users
  (3) localhost:3000/users/1
  (4) localhost:3000/get/users

19. GET으로 'localhost:3000/users/1'을 호출한 결과는 무엇인가?
  (1) User not found
  (2) 신규유저가 추가됨
  (3) [{ "id": 1, "name": "Kim" },{ "id": 2, "name": "Lee" }]
  (4) { "id": 1, "name": "Kim" }

20. DELETE로 'localhost:3000/users/1'을 호출한 후, GET으로 'localhost:3000/users/1'을 호출한 결과는 무엇인가?
  (1) User not found
  (2) [{ "id": 1, "name": "Kim" },{ "id": 2, "name": "Lee" }]
  (3) { "id": 1, "name": "Kim" }
  (4) { "id": 2, "name": "Lee" }