23S-COM SCI-130-LEC-1 Final

CHARLES ZHANG

TOTAL POINTS

92 / 100

QUESTION 1

1 1.1 QPS **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

QUESTION 2

2 1.2 Network **1 / 1**

    ✓ **+ 1 pts** *black*

    + 0 pts white

    + 0 pts blank

QUESTION 3

3 1.3 Time **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

    + 0 pts blank

QUESTION 4

4 1.4 Static File Handler **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

QUESTION 5

5 1.5 In-process queue **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

QUESTION 6

6 1.6 HTTP error code **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

QUESTION 7

7 1.7 Disk free **1 / 1**

    ✓ **+ 1 pts** *black*

    + 0 pts white

    + 0 pts blank

QUESTION 8

8 1.8 Invalid file accesses **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

QUESTION 9

9 1.9 System memory free **1 / 1**

    ✓ **+ 1 pts** *black*

    + 0 pts white

    + 0 pts blank

QUESTION 10

10 1.10 In-process cache size **1 / 1**

    + 0 pts black

    ✓ **+ 1 pts** *white*

    + 0 pts blank

QUESTION 11

11 1.11 CPU usage **1 / 1**

    ✓ **+ 1 pts** *black*

+ 0 pts white

12 1.12 Number of threads in reserved pool **0 / 1**

✓ **+ 0 pts** *black*

  + 1 pts white

  + 0 pts blank

13 2a **3 / 3**

  + 0 pts A

  + 0 pts B

✓ **+ 3 pts** *C*

14 2b **3 / 3**

  + 0 pts A

  + 0 pts B

✓ **+ 3 pts** *C*

  + 0 pts D

15 3 **3 / 3**

  + 0 pts A

  + 0 pts B

✓ **+ 3 pts** *C*

16 4 **6 / 10**

✓ **+ 2 pts** *D .* A*

  + 2 pts A .* E

✓ **+ 2 pts** *E .* B*

  + 2 pts B .* F

✓ **+ 2 pts** *F .* C*

17 5.1 **2 / 2**

✓ **+ 2 pts** *2*

  + 0 pts Incorrect

  + 0 pts Blank

18 5.2 **2 / 2**

✓ **+ 2 pts** *1*

  + 0 pts Incorrect

  + 0 pts Blank

19 5.3 **2 / 2**

✓ **+ 2 pts** *3*

  + 0 pts Incorrect

  + 0 pts Blank

20 6a **2 / 2**

  + 0 pts A

✓ **+ 2 pts** *B*

21 6b **2 / 2**

  - 1 pts A

  - 1 pts B

  + 2 pts C

✓ **+ 2 pts** *D*

22 6c **3 / 3**

  + 0 pts A

  + 0 pts B

✓ **+ 3 pts** *C*

23 7 **8 / 8**

   + 2 pts Source control

✓ **+ 2 pts** *Code review*

✓ **+ 2 pts** *Testing*

✓ **+ 2 pts** *Continuous build/integration*

✓ **+ 2 pts** *Monitoring/alerting*

   + 0 pts 1 Incorrect or duplicate answer

   + 0 pts 2 incorrect or duplicate answers

   + 0 pts Thread safety

   + 2 pts Refactoring

   + 2 pts Design review

   + 2 pts Staged environments

   + 0 pts Readability

   + 2 pts Reuse existing libraries

   + 2 pts Input sanitization

   + 0 pts Logging

   + 2 pts Static/runtime analysis

   + 0 pts Dependency Injection

   + 2 pts Postmortems

   + 2 pts Communication

   + 0 pts Documentation

   + 2 pts Containerization

   + 0 pts Standerdized dev environment

   + 0 pts Small commits

   + 2 pts QoS channel

24 8 **4 / 4**

✓ **+ 2 pts** *Represents several concepts*

   + 2 pts Kitchen sink methods

✓ **+ 2 pts** *Annoying error handling*

   + 2 pts Requires careful orchestration by the caller

   + 2 pts Small changes in usage result in unexpected large changes in behavior

   + 2 pts Stringly-typed

   + 0 pts Incorrect or duplicate answer

   + 2 pts Exposes implementation details

   + 2 pts Easy to misuse

   + 2 pts Non-extensible

   + 0 pts Unreliable

   + 2 pts Works across different layers of abstraction

   + 0 pts Bad naming

   + 0 pts Vague

   + 0 pts Poor documentation

   + 2 pts Multiple ways to do the same thing

25 9 **7 / 10**

   - 1 pts A

✓ **+ 2 pts** *B*

✓ **+ 2 pts** *C*

   - 1 pts D

   - 1 pts E

   - 1 pts F

   + 2 pts G

✓ **- 1 pts** *H*

   - 1 pts I

   - 1 pts J

✓ **+ 2 pts** *K*

   - 1 pts L

✓ **+ 2 pts** *M*

   - 1 pts N

26 10 **14 / 14**

- **1 pts** A

✓ **+ 2 pts** *B*

✓ **+ 2 pts** *C*

- **1 pts** D

- **1 pts** E

- **1 pts** F

- **1 pts** G

✓ **+ 2 pts** *H*

✓ **+ 2 pts** *I*

- **1 pts** J

✓ **+ 2 pts** *K*

✓ **+ 2 pts** *L*

- **1 pts** M

✓ **+ 2 pts** *N*

QUESTION 27

27 **11** **4 / 4**

- **2 pts** A

- **2 pts** B

✓ **+ 4 pts** *C*

- **2 pts** D

- **2 pts** E

+ 0 pts blank

QUESTION 28

28 **12** **16 / 16**

+ 2 pts impl(1-2) Use parameter object

+ 2 pts impl(1-2): Pass strings by const ref

+ 2 pts impl(2): Remove Fake logic from impl

+ 2 pts impl(2) Better parameter name for "f"

✓ **+ 2 pts** *impl(2): Use enum instead of stringly typed updaterType*

✓ **+ 2 pts** *impl(3) Flip comparison*

+ 2 pts impl(4): What does this do?

✓ **+ 2 pts** *impl(6) Bad comment, remove*

✓ **+ 2 pts** *impl(9-17) Unnest if statements*

+ 2 pts impl(10): More logic needed for checking root path?

+ 2 pts impl(10): if(rootPath)

+ 2 pts impl(11) Use parameter object

+ 2 pts impl(11): Return pointer

+ 2 pts impl(11): Rename FileInPlaceUpdater for consistency

+ 2 pts impl(11, 16): Incorrect object instantiation

+ 2 pts impl(12): Add better error handling

+ 2 pts impl(13) Return something better than nullptr

+ 2 pts impl(16): Return pointer

✓ **+ 2 pts** *impl(21) Unnecessary helper method*

+ 2 pts impl(28) Don't throw exception in C++ (only if impl(21) comment is missing)

+ 2 pts test(1, 10) Bad comments, remove

+ 2 pts test(2): Use text fixture

+ 2 pts test(2): Bad variable names

+ 2 pts test(2): Factory should be first parameter

+ 2 pts test(4) Change to test state instead of test interaction

+ 2 pts test(4-5): EXPECT_CALL needs to be after the call

✓ **+ 2 pts** *test(5): prepareForFake() should be called exactly once*

+ 2 pts test(7): parameter not pointer

+ 2 pts test(8): Need to delete after new

+ 2 pts test(11): Keep test suite names consistent

✓ **+ 2 pts** *test(13) : Missing function argument*

+ 2 pts test(14) Flip comparison order

+ 2 pts test(15): Add tests for error conditions

+ 2 pts test(15): Add test for update success

+ 2 pts Code unclear

+ 2 pts Add logging

+ 2 pts impl(4): Null check for f

- 1 pts 1 incorrect comment

- 2 pts 2 incorrect comments

- 4 pts 4 incorrect comments

+ 2 pts Formatting

+ 2 pts impl(24): Make it an else instead

+ 2 pts Testing incorrect assumption

+ 2 pts Missing CL description

+ 2 pts Missing imports

+ 2 pts test(6): Split into two tests

✓ **+ 2 pts** *Return unique_prt*

+ 2 pts Handle empty root path for

FileUpdaterWithBackup

+ 2 pts impl(10): Use std::string::empty

Name: Charles Zhong   UID: 30541365 9

Answer Sheet

(1) Fill in one box per row:

black	white	
☐	☑	Queries per second processed ✓
☒	☐	Total network bandwidth being used ✓
☐	☑	Average length of time to process a request ✓
☐	☒	Number of requests hitting the StaticFileHandler ✓
☐	☒	Number of requests in an in-process queue waiting to be processed ✓
☐	☑	Fraction of requests with http error code in response ✓
☒	☐	Amount of disk free ✓
☐	☑	Number of invalid file accesses in the ImageFileHandler ✓
☑	☐	Amount of system memory free ✓
☐	☑	Amount of memory used by an in-process cache ✓
☒	☐	Amount of CPU in use ✓
☒	☐	Number of threads remaining in a reserved thread pool ✓

(2a) Fill in one:
☐ A ☐ B ☒ C

(2b) Fill in one:
☐ A ☐ B ☒ C ☐ D

(3) Fill in one:
☐ A ☐ B ☒ C

(4) Write comma-separated letters:
D, E, A, F, B, C

(5) Enter the number for each letter:
A: 2   B: 1   C: 3

(6a) Fill in one:
☐ A ☒ B

(6b) Fill in one:
☐ A ☐ B ☐ C ☒ D

(6c) Fill in one:
☐ A ☐ B ☒ C

Name: Charles Zhang   UID: JOS413655

(7) 3 words max for each

1. Unit/integration testing

2. Code reviews

3. Monitoring/alerting

4. Multiple deployments (local Docker before GCP Docker)

5. CI/CD

(8) 1 sentence max for each property

1. Awkward to use (i.e, bad error handling, small changes result in large implementation changes, etc. ↓input

2. Tries to do too many things

(9) Fill in all that apply:

☐ ■ ■ ☐ ☐ ☐ ☐ ■ ☐ ☐ ■ ☐ ■ ☐
A B C D E F G H I J K L M N

(10) Fill in all that apply:

☐ ■ ■ ☐ ☐ ☐ ☐ ■ ■ ☐ ■ ■ ☐ ■
A B C D E F G H I J K L M N

(11) Fill in all that apply:

☐ ☐ ■ ☐ ☐
A B C D E

Name: Charles Zhang   UID: 305413659

(12) For "filename", indicate either "impl" or "test"
- Be sure to address both files.
- We are looking for no more than 8 good comments; recall that too many is also bad practice.

Filename +Line	Comment
impl 1-2	Can you use a more descriptive type than std·string for your params?
impl 3	Can you flip this conditional so the constant is on the RHS for readability?
impl 9-15	Could you un-nest this statement? The if/else are too far apart
impl 21-29	This shouldn't be a separate function, all it does is return the value of the boolean
impl 2?-2?	
test 5	Why is this at least 1 time? Shouldn't it always be exactly 1 time?
test 13	This function call is missing an input param. If f only exists for "FakePointers", maybe rethink this function definition
impl 1	Can you wrap this pointer return type in a smart pointer? This could also fix the memory leak in the test file
impl 6	Unnecessary comment