Software and Data Engineering
CSL7090
Quiz 1

1. A 6-person startup must ship three big features in 12 weeks. Requirements are evolving weekly; deployments happen daily; outages must be minimal but global scale is not required yet. Which architecture is MOST appropriate now?
   A. Microservices with per-service databases
   B. Modular monolith with clear module boundaries
   C. Pure P2P across mobile clients
   D. SOA with enterprise service bus

   Answer: B

2. A mighty foundation, a layer below,
   Where data and logic are ready to grow.
   Upon this foundation, the next layer stands,
   Handling the business with thoughtful commands.
   Then the top layer, a user's delight,
   Presents all the data, so shiny and bright.

   What pattern am I**?**

   A) Monolithic
   B) Client-Server
   C) Layered
   D) Pipes and Filters

   Answer: C

3. An ETL pipeline: **CSV → Validate → Normalize → Enrich → Aggregate → Parquet**. A new upstream source sends JSON instead of CSV. To keep filters reusable and loosely coupled, where should format conversion primarily live?
   A. Inside each filter that touches input
   B. In the pipes connecting filters
   C. In the enrich filter only
   D. In a new downstream report service

   Answer: B

4. A kingdom divided, a realm torn apart,
   Each piece a small kingdom, a separate art.
   They speak a new language, a light, simple call,
   To build a great service, and stand up so tall.
   What pattern am I?

A) Layered
B) Event-driven
C) Microservices
D) Monolithic

Answer: C

5. A fintech adds real-time AML checks that increase CPU and I/O. Peak p95 latency goes from 250ms to 320ms, but fraud losses drop 40%. Which trade-off is most defensible?
   A. Remove encryption on internal hops
   B. Accept latency for stronger security controls
   C. Use eventual consistency for money transfers
   D. Push AML to client devices

   Answer: B

6. A startup has 6 engineers, daily deploys, evolving requirements, and a 10-week deadline. Which architecture optimizes delivery without killing maintainability?
   A. Microservices with per-service DBs now
   B. Modular monolith with clear boundaries
   C. Pure P2P across user devices
   D. ESB-centric SOA

   Answer: B

7. You're revising a design doc to separate **architecture** from **design** decisions. Which belong in **architecture**?
   A. Pick REST vs GraphQL for inter-service APIs
   B. Choose Postgres vs Mongo (primary store)
   C. Select JWT claim set for a specific endpoint
   D. Decide monolith vs microservices

   Answer: A, B, D

8. A gaming backend considers lock-free, highly optimized code paths. What are the architectural risks?
   A. Reduced maintainability for perf gains
   B. Harder onboarding; fewer developers can safely modify
   C. Guaranteed elimination of concurrency bugs
   D. Potential for subtle starvation or fairness issues

   Answer: A, B, D

9. A bank mandates strong consistency for balances and transfers. Which scalability lever is least applicable?

A. Sharding read-only replicas for statements
B. CQRS with write-side strict consistency
C. Eventual consistency for transfer ledger
D. Horizontal app tier scaling

Answer: C

10. Which arguments justify **server affinity** behind a broker?
A. Cache locality for repeated user flows
B. Consistent session state without external store
C. Eliminates the need for health checks
D. Lower average latency for stateful flows

Answer: A, B, D

11. You're migrating a thick desktop client to **client-server**. Benefits?
A. Centralized updates & patching
B. Single source of truth for data
C. Guaranteed offline functionality
D. Reduced client hardware specs

Answer: A, B, D

12. Choosing **broker** vs **master-slave** controller: which statements hold?
A. Broker routes; servers remain autonomous
B. Master assigns tasks with authoritative control
C. Broker equals forced synchronous execution
D. Master-slave can centralize scheduling logic

Answer: A, B, D

13. A global news site uses two public brokers with 20 app servers. Clients only know one DNS name. Which advantage is most central to the broker idea here?
A. Clients must know all server IPs
B. New servers only register with broker
C. Clients maintain server maps locally
D. Brokers contain all business rules

Answer: B

14. Signals you're drifting toward a **distributed monolith**:
A. Many services share one schema
B. A change in one service requires coordinated deploys across several
C. Independent CI/CD per service with backward-compatible APIs
D. Network chatter is high for simple user interactions

Answer: A, B, D

15. Operational benefits of **pipes & filters** include:
    A. Easy pinpointing of bottleneck filters
    B. Replacement of a single filter without pipeline redesign
    C. Filters must know internals of all neighbors
    D. Natural fit for ETL & staged validations

    Answer: A, B, D

16. When would you **prefer P2P** for distribution?
    A. Massive popular file that gets faster as more peers join
    B. Need for strict audit trail & compliance
    C. Desire to minimize central bandwidth costs
    D. Requirement for global takedown control

    Answer: A, C

17. You’re building a personalized alerts system for sports and finance. **Pub-Sub** helps because…
    A. Publishers don’t need to know who subscribes
    B. Topic routing fans out to many subscribers
    C. Message broker can scale separately
    D. Subscribers must modify publisher content directly

    Answer: A, B, C

18. What are the risks with **pub-sub** at scale:
    A. Broker as a potential single point of failure
    B. Exactly-once delivery is trivial
    C. Harder tracing of async message flows
    D. Potential duplication/out-of-order messages

    Answer: A, C, D

19. A host of equals, with no one in charge,
    They share all their data, their network is large.
    Each node can both listen and send with great speed,
    Helping each other in moments of need.
    What pattern am I?
    A) Client-Server
    B) Service-Oriented
    C) Peer-to-Peer
    D) Monolithic

    Answer: C

20. A data platform chooses shared object storage for raw data; multiple apps read (no writes). Architecturally this converges toward:
 A. Pub-Sub-like read-only consumption
 B. Master-slave control
 C. P2P discovery
 D. MVC

Answer: A

21. Service Bus vs Pub-Sub: pick the accurate contrasts.
 A. Bus is passive; consumers poll/browse offerings
 B. Pub-Sub broker actively pushes to subscribers
 C. Bus provides privacy by default between publishers/subscribers
 D. Pub-Sub supports interest-based selective delivery

Answer: B, D

22. Operational concerns for shared data under heavy write contention:
 A. Deadlocks and lock contention
 B. Transaction isolation level tuning
 C. Guaranteed linearizability without trade-offs
 D. Hotspot indexing and partitioning

Answer: A, B, D

23. You add **retry + timeout** tactics to a payment API to improve availability during transient faults. Which architectural side effects must you analyze?
 A. Increased downstream load amplification during incidents
 B. Lower tail latency in all failure modes
 C. Risk of duplicate effects on non-idempotent operations
 D. Interactions with circuit breakers and backoff policies

Answer: A, C, D

24. A distributed platform wants location-independent service access and fault isolation between clients and servers. Which pattern best fits?
 A. Interpreter
 B. Broker (decoupled remote service invocations; request forwarding & results/exceptions handling)
 C. Event-bus
 D. Blackboard

Answer: B

25. A single commander awaits a command,
To process a query and lend a helping hand.

It's ready to listen, a server so grand,
While clients request, from all over the land.
What pattern am I?
A) Peer-to-Peer
B) Client-Server
C) Message Bus
D) Microkernel

Answer: B

26. A fraud team needs to combine outputs from OCR, rules, heuristics, and ML to converge on a case decision with partial knowledge at each step. Which pattern is most natural?
 A. Blackboard
 B. Pipes-and-filters
 C. Client-server
 D. Master-slave

Answer: A

27. Product managers want business users to author expressions like:

```
if customer.segment in ["Gold","Platinum"] and order.total >
10_000 then requireManualReview()
```

You need an architecture that cleanly executes domain expressions without recompiling the app. Choose:
 A. Event-bus
 B. Interpreter
 C. Peer-to-peer
 D. MVC

Answer: B

28. When is **master-slave** (controller–workers) more appropriate than a broker?
 A. You need centralized scheduling and progress control
 B. Workers must follow assigned sub-tasks deterministically
 C. Services should remain fully autonomous and only be “routed to”
 D. The coordinator must split jobs and aggregate results

Answer: A, B, D

29. You want **centralized updates, access control, and a single source of truth** for records used by many users simultaneously. Which paradigm is the natural baseline?
 A. Client-server
 B. P2P

C. Interpreter
D. Blackboard

Answer: A

30. Choosing **microservices** over a monolith introduces which governance risks if not countered?
 A. Distributed monolith via shared DB schema
 B. Chatty interfaces inflating p99 latency
 C. Easier independent deploys with backward-compatible APIs
 D. Complex tracing/observability needs for async paths

 Answer: A, B, D