Lesson 6.2: Core Banking and Legacy Systems – Quiz

Prof. Dr. Joerg Osterrieder

Question 1

What does "CORE" stand for in core banking?

A. Central Operations and Regulatory Engine
B. Centralized Online Real-time Exchange
C. Customer-Oriented Revenue Environment
D. Consolidated Operational Risk Engine

Question 1

What does "CORE" stand for in core banking?

A. Central Operations and Regulatory Engine
B. Centralized Online Real-time Exchange
C. Customer-Oriented Revenue Environment
D. Consolidated Operational Risk Engine

Answer: B
CORE = Centralized Online Real-time Exchange. It refers to the central software platform that processes daily banking transactions and posts updates to accounts in real time.

Question 2

Which fundamental accounting principle does a general ledger enforce?

A. Single-entry bookkeeping: each transaction recorded once
B. Double-entry bookkeeping: every debit has a matching credit
C. Cash-basis accounting: transactions recorded when cash changes hands
D. Mark-to-market: assets valued at current market price daily

Question 2

Which fundamental accounting principle does a general ledger enforce?

A. Single-entry bookkeeping: each transaction recorded once

B. Double-entry bookkeeping: every debit has a matching credit

C. Cash-basis accounting: transactions recorded when cash changes hands

D. Mark-to-market: assets valued at current market price daily

Answer: B

The general ledger enforces double-entry bookkeeping, ensuring Assets = Liabilities + Equity at all times. Every financial transaction has offsetting debit and credit entries.

Question 3

What percentage of ATM transactions globally still run on COBOL?

A. 25%
B. 50%
C. 75%
D. 95%

Question 3

What percentage of ATM transactions globally still run on COBOL?

A. 25%

B. 50%

C. 75%

D. 95%

Answer: D

COBOL, first developed in 1959, still processes approximately 95% of ATM transactions and 80% of in-person transactions worldwide, with 220 billion lines of COBOL in production today.

Question 4

What is the primary risk of COBOL-based legacy systems?

A. COBOL is too slow for modern transaction volumes
B. The shrinking talent pool of COBOL programmers and undocumented business logic
C. COBOL cannot handle multi-currency transactions
D. Mainframes are physically too large for modern data centers

Question 4

What is the primary risk of COBOL-based legacy systems?

A. COBOL is too slow for modern transaction volumes

B. The shrinking talent pool of COBOL programmers and undocumented business logic

C. COBOL cannot handle multi-currency transactions

D. Mainframes are physically too large for modern data centers

Answer: B

The language itself performs well (99.999% uptime). The risk is the shrinking pool of programmers (average age 55–60) and decades of undocumented business logic embedded in the code.

Question 5

Technical debt is best described as:

A. The cost of purchasing software licenses for legacy systems
B. The implied cost of future rework from choosing quick solutions over better long-term approaches
C. The depreciation of IT hardware over time
D. The financial debt a bank takes on to fund IT projects

Question 5

Technical debt is best described as:

A. The cost of purchasing software licenses for legacy systems

B. The implied cost of future rework from choosing quick solutions over better long-term approaches

C. The depreciation of IT hardware over time

D. The financial debt a bank takes on to fund IT projects

Answer: B

Technical debt is a metaphor coined by Ward Cunningham (1992). Like financial debt, you pay "interest" in the form of slower development, more bugs, and higher maintenance costs.

Question 6

What proportion of their IT budget do most traditional banks spend on maintenance ("run the bank") rather than innovation?

A. 20–30%
B. 40–50%
C. 70–80%
D. 95–100%

Question 6

What proportion of their IT budget do most traditional banks spend on maintenance ("run the bank") rather than innovation?

A. 20–30%
B. 40–50%
C. 70–80%
D. 95–100%

Answer: C
Most traditional banks spend 70–80% of IT budgets on maintaining existing systems, leaving only 20–30% for innovation. This is a direct consequence of accumulated technical debt.

Question 7

What happened during TSB Bank's core banking migration in 2018?

A. A successful phased rollout over 2 years with no customer impact
B. A big bang cutover that locked out 1.9 million customers and exposed some to other people's accounts
C. A strangler fig migration that was halted due to regulatory concerns
D. A cloud migration that was reversed after data sovereignty concerns

Question 7

What happened during TSB Bank's core banking migration in 2018?

A. A successful phased rollout over 2 years with no customer impact

B. A big bang cutover that locked out 1.9 million customers and exposed some to other people's accounts

C. A strangler fig migration that was halted due to regulatory concerns

D. A cloud migration that was reversed after data sovereignty concerns

Answer: B

TSB attempted a big bang migration from Lloyds' legacy platform to Sabadell's Proteo4UK. The failed cutover cost £330M, caused the CEO to resign, and 80,000 customers left the bank.

Question 8

The strangler fig pattern for software migration is named after:

A. A computer science algorithm for graph traversal
B. A tropical vine that gradually grows around and replaces a host tree
C. An economic theory about disruptive innovation
D. A database migration technique invented by IBM

Question 8

The strangler fig pattern for software migration is named after:

A. A computer science algorithm for graph traversal

B. A tropical vine that gradually grows around and replaces a host tree

C. An economic theory about disruptive innovation

D. A database migration technique invented by IBM

Answer: B

The strangler fig pattern, popularized by Martin Fowler (2004), is named after the strangler fig tree. The vine gradually grows around a host tree, eventually replacing it—just as new services gradually replace legacy components.

Question 9

Which migration strategy involves running old and new systems simultaneously, comparing outputs daily?

A. Big bang
B. Strangler fig
C. Parallel run
D. Canary deployment

Question 9

Which migration strategy involves running old and new systems simultaneously, comparing outputs daily?

A. Big bang
B. Strangler fig
C. Parallel run
D. Canary deployment

Answer: C
Parallel run operates both systems simultaneously and compares results. It offers maximum safety but at double the operating cost and with significant data synchronization challenges.

Question 10

Which of these is a banking-specific challenge for cloud migration?

A. Cloud providers do not offer sufficient compute capacity
B. Data sovereignty regulations require customer data to stay in specific jurisdictions
C. Cloud platforms cannot support relational databases
D. Cloud-native applications cannot achieve regulatory compliance

Question 10

Which of these is a banking-specific challenge for cloud migration?

A. Cloud providers do not offer sufficient compute capacity

B. Data sovereignty regulations require customer data to stay in specific jurisdictions

C. Cloud platforms cannot support relational databases

D. Cloud-native applications cannot achieve regulatory compliance

Answer: B

Data sovereignty (GDPR, FINMA, and similar regulations) requires that customer data remains within specific geographic boundaries. Banks must select cloud regions carefully and may need multi-cloud or hybrid strategies.

Question 11

Approximately what percentage of banking workloads were in public cloud as of 2024?

A. Less than 5%
B. About 15%
C. About 50%
D. Over 80%

Question 11

Approximately what percentage of banking workloads were in public cloud as of 2024?

A. Less than 5%

B. About 15%

C. About 50%

D. Over 80%

Answer: B

Only approximately 15% of banking workloads are in public cloud. Hybrid cloud dominates: sensitive data and core processing remain on-premise while analytics and development environments move to cloud.

Question 12

In a microservice architecture, how do individual services typically communicate?

A. Through a shared database that all services read from and write to
B. Via APIs (synchronous) or message queues (asynchronous)
C. By sharing memory within a single process
D. Through direct file system access on a shared server

Question 12

In a microservice architecture, how do individual services typically communicate?

A. Through a shared database that all services read from and write to

B. Via APIs (synchronous) or message queues (asynchronous)

C. By sharing memory within a single process

D. Through direct file system access on a shared server

Answer: B

Microservices communicate via APIs (typically REST or gRPC for synchronous calls) or message queues/event streams (Kafka, RabbitMQ for asynchronous communication). Each service owns its own data store.

Question 13

What is a key advantage of monolithic architecture over microservices?

A. Independent deployment of components
B. Better fault isolation between modules
C. Simpler to develop, debug, and deploy initially
D. Superior horizontal scalability

Question 13

What is a key advantage of monolithic architecture over microservices?

A. Independent deployment of components
B. Better fault isolation between modules
C. Simpler to develop, debug, and deploy initially
D. Superior horizontal scalability

Answer: C
Monolithic applications are simpler to develop, debug (single process), and deploy initially. Microservices introduce distributed-system complexity (network failures, eventual consistency, service discovery) that is only justified at scale.

Question 14

What does "API-first" design mean?

A. APIs are built after the application is complete, as an afterthought
B. The API contract is designed before the implementation, and the API is treated as the product
C. Only external partners can access the system through APIs
D. APIs are prioritized over the user interface in the development schedule

Question 14

What does "API-first" design mean?

A. APIs are built after the application is complete, as an afterthought

B. The API contract is designed before the implementation, and the API is treated as the product

C. Only external partners can access the system through APIs

D. APIs are prioritized over the user interface in the development schedule

Answer: B

API-first means designing the API contract (using specifications like OpenAPI/Swagger) before building the implementation. This enables parallel development and ensures consistent, well-documented interfaces.

Question 15

Apache Kafka is primarily used for:

A. Relational database management
B. Distributed event streaming and real-time data processing
C. Static web page hosting
D. Batch file transfer between banking systems

Question 15

Apache Kafka is primarily used for:

A. Relational database management

B. Distributed event streaming and real-time data processing

C. Static web page hosting

D. Batch file transfer between banking systems

Answer: B

Apache Kafka is a distributed event streaming platform used for building real-time data pipelines. In banking, it enables real-time fraud detection, instant balance updates, and regulatory event reporting.

Question 16

What is CQRS (Command Query Responsibility Segregation)?

A. A database backup strategy that creates read-only replicas
B. A pattern that separates the data model for reading (queries) from the model for writing (commands)
C. A regulatory requirement for separating customer data from operational data
D. A cloud migration strategy that processes reads before writes

Question 16

What is CQRS (Command Query Responsibility Segregation)?

A. A database backup strategy that creates read-only replicas

B. A pattern that separates the data model for reading (queries) from the model for writing (commands)

C. A regulatory requirement for separating customer data from operational data

D. A cloud migration strategy that processes reads before writes

Answer: B

CQRS separates the read model (optimized for queries) from the write model (optimized for commands). This allows each side to scale independently and use different data stores optimized for their workload.

Question 17

What is the key difference between a data lake and a data warehouse?

A. A data lake uses schema-on-read; a data warehouse uses schema-on-write
B. A data lake can only store structured data; a data warehouse stores unstructured data
C. A data lake is always more expensive than a data warehouse
D. A data warehouse stores raw data; a data lake stores processed data

Question 17

What is the key difference between a data lake and a data warehouse?

A. A data lake uses schema-on-read; a data warehouse uses schema-on-write

B. A data lake can only store structured data; a data warehouse stores unstructured data

C. A data lake is always more expensive than a data warehouse

D. A data warehouse stores raw data; a data lake stores processed data

Answer: A

A data lake ingests raw data without requiring a predefined schema (schema-on-read). A data warehouse requires data to be transformed and structured before loading (schema-on-write). Lakes are more flexible but risk becoming "data swamps" without governance.

Question 18

In data mesh architecture, who owns the data?

A. A centralized data engineering team
B. The Chief Data Officer exclusively
C. Domain teams who treat their data as a product
D. External cloud providers who host the data

Question 18

In data mesh architecture, who owns the data?

A. A centralized data engineering team
B. The Chief Data Officer exclusively
C. Domain teams who treat their data as a product
D. External cloud providers who host the data

Answer: C
Data mesh decentralizes data ownership to domain teams (e.g., payments, lending, compliance). Each team owns and maintains its data as a product, with a federated governance model ensuring interoperability.

Question 19

What percentage of core banking migrations exceed their original timeline, according to industry data?

A. 20%
B. 40%
C. 70%
D. 95%

Question 19

What percentage of core banking migrations exceed their original timeline, according to industry data?

A. 20%
B. 40%
C. 70%
D. 95%

Answer: C
Approximately 70% of core banking migrations exceed their original timeline. Average duration for a large bank is 3–7 years. Success rates are higher with phased approaches (strangler fig) and early regulator engagement.

Question 20

A bank is evaluating cloud-native core banking vendors. Which factor is LEAST relevant to the decision?

A. Vendor's financial stability and long-term viability
B. Regulatory approval for the target jurisdiction
C. The programming language the vendor's engineers prefer personally
D. Data sovereignty and geographic hosting options

Question 20

A bank is evaluating cloud-native core banking vendors. Which factor is LEAST relevant to the decision?

A. Vendor's financial stability and long-term viability

B. Regulatory approval for the target jurisdiction

C. The programming language the vendor's engineers prefer personally

D. Data sovereignty and geographic hosting options

Answer: C

Vendor evaluation should focus on technology architecture, regulatory fit, commercial terms, vendor viability, and implementation track record. The personal language preference of the vendor's engineers is not a material evaluation criterion.