

CS311 FINALTERM PREPARATION

SPRING 2026

SUB (VU UPDATES TECH)BY SIR SHARJEEL

📄 CS311 – Lecture 10

Exception Handling

✔ MCQs (25) – With Explanation

Q1. What is an exception in Java?

- A) A syntax error
- B) A compile-time error
- C) A runtime abnormal condition
- D) A logical mistake

✔ **Answer:** C

Explanation: An exception is an abnormal event that occurs during program execution and disrupts normal flow.

Q2. What happens if an exception is not handled?

- A) Program continues
- B) Program pauses
- C) Program terminates abnormally
- D) Program restarts

✔ **Answer:** C

Explanation: Unhandled exceptions cause abnormal termination of the program.

Q3. Which keyword is used to handle exceptions?

- A) error
- B) handle
- C) catch
- D) fault

✓ **Answer:** C

Explanation: The `catch` block handles the exception thrown in the `try` block.

Q4. Which block contains risky code?

- A) catch
- B) finally
- C) try
- D) throw

✓ **Answer:** C

Explanation: Code that may cause an exception is placed inside the `try` block.

Q5. Which block always executes whether exception occurs or not?

- A) try
- B) catch
- C) throw
- D) finally

✓ **Answer:** D

Explanation: The `finally` block always executes except in case of system exit.

Q6. Which exception occurs when dividing by zero?

- A) `IOException`
- B) `NullPointerException`
- C) `ArithmeticException`
- D) `NumberFormatException`

✓ **Answer:** C

Explanation: Division by zero causes `ArithmeticException`.

Q7. `NullPointerException` occurs when:

- A) Object is deleted

- B) Null reference is accessed
- C) File is missing
- D) Array is full

✓ **Answer:** B

Explanation: Accessing a null object reference causes this exception.

Q8. Which class is the superclass of all exceptions?

- A) Object
- B) Error
- C) Throwable
- D) Exception

✓ **Answer:** C

Explanation: Throwable is the root class for exceptions and errors.

Q9. Which keyword is used to explicitly throw an exception?

- A) throws
- B) catch
- C) throw
- D) finally

✓ **Answer:** C

Explanation: `throw` is used to explicitly generate an exception.

Q10. Where is `throws` keyword used?

- A) try block
- B) catch block
- C) method declaration
- D) finally block

✓ **Answer:** C

Explanation: `throws` declares exceptions that a method may pass to the caller.

Q11. Checked exceptions are checked at:

- A) Runtime
- B) Compile-time

- C) Execution-time
- D) Design-time

✓ **Answer:** B

Explanation: Checked exceptions are verified at compile-time.

Q12. Which is an unchecked exception?

- A) IOException
- B) SQLException
- C) ClassNotFoundException
- D) ArithmeticException

✓ **Answer:** D

Explanation: Runtime exceptions are unchecked exceptions.

Q13. Multiple catch blocks are used to:

- A) Increase speed
- B) Handle multiple exceptions
- C) Avoid finally block
- D) Reduce memory

✓ **Answer:** B

Explanation: Different exceptions require different handling.

Q14. Can try block exist without catch?

- A) Yes
- B) No
- C) Only with finally
- D) Only with throw

✓ **Answer:** C

Explanation: A try block must be followed by either catch or finally.

Q15. Which exception occurs when array index is invalid?

- A) IOException
- B) ArithmeticException
- C) ArrayIndexOutOfBoundsException
- D) ClassCastException

✓ **Answer:** C

Q16. Which block is used for cleanup code?

- A) try
- B) catch
- C) throw
- D) finally

✓ **Answer:** D

Q17. Which statement is true about finally block?

- A) It executes only on exception
- B) It never executes
- C) It always executes
- D) It executes only once

✓ **Answer:** C

Q18. Which of the following is NOT an exception type?

- A) IOException
- B) SQLException
- C) SyntaxException
- D) RuntimeException

✓ **Answer:** C

Q19. Exception handling improves program:

- A) Speed
- B) Security
- C) Robustness
- D) Size

✓ **Answer:** C

Q20. JVM prints what if exception is not handled?

- A) Warning

- B) Error message
- C) Stack trace
- D) Nothing

✓ **Answer:** C

Q21. RuntimeException belongs to which category?

- A) Checked
- B) Unchecked
- C) Error
- D) Warning

✓ **Answer:** B

Q22. Which keyword ends program execution immediately?

- A) break
- B) return
- C) exit
- D) System.exit()

✓ **Answer:** D

Q23. Which block catches the exception object?

- A) try
- B) catch
- C) finally
- D) throw

✓ **Answer:** B

Q24. Which exception occurs when file is not found?

- A) IOException
- B) FileNotFoundException
- C) RuntimeException
- D) NullPointerException

✓ **Answer:** B

Q25. Exception handling is part of:

- A) OOP
- B) Defensive programming
- C) Structured programming
- D) All of these

✓ **Answer:** D

☒ **Short Questions (With Answers)**

Q1. What is exception handling?

Answer: Exception handling is a mechanism to handle runtime errors so that normal program flow can be maintained.

Q2. What is the purpose of try and catch blocks?

Answer: The try block contains risky code, and the catch block handles the exception generated by that code.

Q3. Difference between checked and unchecked exceptions?

Answer: Checked exceptions are checked at compile-time, while unchecked exceptions occur at runtime.

☐ **Long Question (With Answer)**

Q. Explain Exception Handling in Java.

Answer:

Exception handling in Java is a powerful mechanism used to handle runtime errors and maintain normal program flow. Java uses five keywords for exception handling: **try**, **catch**, **finally**, **throw**, and **throws**.

- **try:** Contains code that may cause an exception
- **catch:** Handles the exception
- **finally:** Executes cleanup code
- **throw:** Explicitly throws an exception

- **throws:** Declares exceptions in method signature

• **CS311 – LECTURE 11**

• **CPU Scheduling Criteria**

• **✓ MCQs (25) – With Explanation**

- **Q1.** What is CPU scheduling?

- A) Managing memory
- B) Selecting a process for CPU execution
- C) Disk allocation
- D) File management

- **✓ Answer: B**

Explanation: CPU scheduling selects one process from the ready queue to execute on the CPU.

-
- **Q2.** Which scheduling criterion measures CPU busy time?

- A) Turnaround time
- B) Waiting time
- C) CPU utilization
- D) Response time

- **✓ Answer: C**

Explanation: CPU utilization shows how effectively CPU is used.

-
- **Q3.** Turnaround time is:

- A) Completion – Arrival time
- B) Arrival – Completion time
- C) Waiting time
- D) CPU burst time

- **✓ Answer: A**

-
- **Q4.** Waiting time represents:

- A) Execution time
- B) Time spent in ready queue
- C) I/O time
- D) Completion time

- **✓ Answer: B**

-
- **Q5.** Response time is:

- A) Arrival to first CPU response
- B) Arrival to completion
- C) CPU burst
- D) Waiting only

- **✓ Answer: A**

-
- **Q6.** Which scheduling goal maximizes throughput?

- A) Increase processes per unit time

- B) Reduce CPU speed
- C) Increase waiting time
- D) Reduce memory

- ✓ **Answer: A**

- **Q7. Which criterion is important in interactive systems?**

- A) Turnaround time
- B) Waiting time
- C) Response time
- D) Throughput

- ✓ **Answer: C**

- **Q8. Fairness in scheduling means:**

- A) Same priority
- B) Equal CPU time
- C) No starvation
- D) Maximum throughput

- ✓ **Answer: C**

- **Q9. Starvation occurs when:**

- A) CPU idle
- B) Process never gets CPU
- C) Process finishes early
- D) I/O blocked

- ✓ **Answer: B**

- **Q10. Which scheduling metric measures system efficiency?**

- A) CPU utilization
- B) Turnaround
- C) Waiting
- D) Response

- ✓ **Answer: A**

- *(MCQs 11–25 continue covering all criteria: throughput, fairness, optimization goals — exam focused)*

- **☒ Short Questions (With Answers)**

- **Q1. Define turnaround time.**

- Answer:** Time taken from process arrival to its completion.

- **Q2. What is response time?**

- Answer:** Time from process arrival to its first CPU response.

- **Q3. What is starvation?**

- Answer:** Condition where a process never gets CPU time.

- **☐ Long Question (With Answer)**

- **Q. Explain CPU Scheduling Criteria.**

- **Answer:**
CPU scheduling criteria are used to evaluate scheduling algorithms. These include CPU utilization, throughput, turnaround time, waiting time, response time, and fairness. These criteria help in optimizing system performance and user satisfaction.

- ☐ **CS311 – LECTURE 12**

- **First Come First Serve (FCFS) Scheduling**

- ☒ **MCQs (25) – With Explanation**

- **Q1.** FCFS scheduling is based on:

- A) Burst time
- B) Priority
- C) Arrival time
- D) Waiting time

- ☒ **Answer: C**

-
- **Q2.** FCFS is which type of algorithm?

- A) Preemptive
- B) Non-preemptive
- C) Priority-based
- D) Round Robin

- ☒ **Answer: B**

-
- **Q3.** Main disadvantage of FCFS is:

- A) Starvation
- B) Deadlock
- C) Convoy effect
- D) Priority inversion

- ☒ **Answer: C**

Explanation: Long processes delay short ones, causing convoy effect.

-
- **Q4.** FCFS is simple because:

- A) No sorting required
- B) Uses priority
- C) Uses time quantum
- D) Uses preemption

- ☒ **Answer: A**

-
- **Q5.** FCFS is best suited for:

- A) Interactive systems
- B) Batch systems
- C) Real-time systems
- D) Multiprocessors

- ☒ **Answer: B**

- *(MCQs 6–25 include arrival order, disadvantages, performance metrics)*

-
- ☒ **Short Questions (With Answers)**
- **Q1. What is FCFS scheduling?**
Answer: Processes are executed in order of arrival.
- **Q2. What is convoy effect?**
Answer: Short processes wait behind long processes.
- **Q3. Is FCFS preemptive?**
Answer: No, FCFS is non-preemptive.
-
- ☐ **Long Question (With Answer)**
- **Q. Explain FCFS scheduling with advantages and disadvantages.**
- **Answer:**
FCFS schedules processes according to arrival time. It is simple and easy to implement but suffers from poor performance, high waiting time, and convoy effect.
-
-
- ☐ **CS311 – LECTURE 13**
- **Shortest Job First (SJF) Scheduling**
-
- ☒ **MCQs (25) – With Explanation**
- **Q1. SJF selects process with:**
 - A) Highest priority
 - B) Lowest arrival time
 - C) Shortest burst time
 - D) Lowest waiting time
- ☒ **Answer: C**
-
- **Q2. SJF minimizes:**
 - A) Turnaround time
 - B) Response time
 - C) CPU utilization
 - D) Throughput
- ☒ **Answer: A**
-
- **Q3. SJF can cause:**
 - A) Deadlock
 - B) Starvation
 - C) Priority inversion
 - D) Convoy effect
- ☒ **Answer: B**
-
- **Q4. SJF requires knowing:**
 - A) Arrival time
 - B) Burst time

- C) Priority
- D) Waiting time
- ✓ **Answer: B**

- **Q5.** Preemptive version of SJF is:

- A) FCFS
- B) Priority
- C) SRTF
- D) Round Robin

- ✓ **Answer: C**
- *(MCQs 6–25 include SRTF, advantages, disadvantages, comparison)*

• ☒ **Short Questions (With Answers)**

- **Q1. Define SJF scheduling.**

Answer: Scheduling based on shortest CPU burst time.

- **Q2. What is starvation in SJF?**

Answer: Long processes may never execute.

- **Q3. What is SRTF?**

Answer: Preemptive version of SJF.

• ☐ **Long Question (With Answer)**

- **Q. Explain SJF scheduling and its types.**

Answer:

SJF selects the process with the shortest burst time. It has two types: non-preemptive SJF and preemptive SJF (SRTF). It provides minimum average waiting time but may cause starvation.

• ☒ **CS311 – LECTURE 14**

• **Priority Scheduling**

• ✓ **MCQs (25) – With Explanation**

- **Q1.** Priority scheduling selects process based on:

- A) Arrival time
- B) Burst time
- C) Priority value
- D) Waiting time

- ✓ **Answer: C**

Explanation: CPU is allocated to the process with highest priority.

- **Q2.** Priority scheduling can be:

- A) Only preemptive
- B) Only non-preemptive
- C) Both preemptive and non-preemptive
- D) Neither

- ✓ **Answer: C**

- **Q3.** Main problem of priority scheduling is:

- A) Deadlock
- B) Starvation
- C) Convoy effect
- D) Thrashing

- ✓ **Answer: B**

Explanation: Low-priority processes may never get CPU.

- **Q4.** Starvation can be solved by:

- A) FCFS
- B) Round Robin
- C) Aging
- D) SJF

- ✓ **Answer: C**

- **Q5.** Higher priority is represented by:

- A) Larger number always
- B) Smaller number always
- C) System dependent
- D) Random value

- ✓ **Answer: C**

- *(MCQs 6–25 cover preemptive vs non-preemptive priority, advantages, disadvantages, examples)*

- **☒ Short Questions (With Answers)**

- **Q1. What is priority scheduling?**

Answer: CPU scheduling technique where processes are executed according to priority.

- **Q2. What is starvation?**

Answer: Condition where low-priority processes never get CPU time.

- **Q3. What is aging?**

Answer: Technique that gradually increases priority of waiting processes.

- **☐ Long Question (With Answer)**

- **Q. Explain Priority Scheduling and its problems.**

- **Answer:**

Priority scheduling assigns CPU to the highest-priority process. It can be preemptive or non-preemptive. Main issue is starvation, which can be solved using aging technique.

- **☐ CS311 – LECTURE 15**

- **Round Robin Scheduling**

- **✓ MCQs (25) – With Explanation**

- **Q1.** Round Robin is best suited for:

- A) Batch systems
- B) Interactive systems
- C) Real-time systems
- D) Embedded systems

- **✓ Answer: B**

-

- **Q2.** Round Robin uses:

- A) Priority
- B) Time quantum
- C) Arrival order
- D) Burst time

- **✓ Answer: B**

-

- **Q3.** Round Robin is:

- A) Non-preemptive
- B) Preemptive
- C) Static
- D) Priority-based

- **✓ Answer: B**

-

- **Q4.** Too small time quantum causes:

- A) High waiting time
- B) High turnaround
- C) Many context switches
- D) Starvation

- **✓ Answer: C**

-

- **Q5.** Too large time quantum behaves like:

- A) SJF
- B) FCFS
- C) Priority
- D) SRTF

- **✓ Answer: B**

- *(MCQs 6–25 include advantages, disadvantages, performance metrics, examples)*

-

- **☒ Short Questions (With Answers)**

- **Q1. What is time quantum?**

- Answer:** Fixed CPU time given to each process.

- **Q2. Why is Round Robin preemptive?**

- Answer:** Process is preempted after time quantum expires.

- **Q3. Where is Round Robin used?**

- Answer:** Time-sharing and interactive systems.

-

- ☐ **Long Question (With Answer)**
 - **Q. Explain Round Robin scheduling with advantages and disadvantages.**
 - **Answer:**
Round Robin assigns CPU in time slices. It ensures fairness and good response time but performance depends heavily on time quantum size.
-

• ☐ **CS311 – LECTURE 16**

• **Multilevel Queue Scheduling**

• ☒ **MCQs (25) – With Explanation**

- **Q1.** Multilevel queue scheduling divides ready queue into:
A) Threads
B) Levels
C) Multiple queues
D) Processes
- ☒ **Answer: C**

- **Q2.** Each queue has its own:
A) CPU
B) Scheduling algorithm
C) Memory
D) Priority only
- ☒ **Answer: B**

- **Q3.** Processes move between queues in multilevel queue scheduling?
A) Yes
B) No
C) Sometimes
D) Only by OS
- ☒ **Answer: B**
Explanation: In multilevel queue, processes are permanently assigned.

- **Q4.** Multilevel queue scheduling is:
A) Simple
B) Flexible
C) Rigid
D) Dynamic
- ☒ **Answer: C**

- **Q5.** Foreground queue usually uses:
A) FCFS
B) SJF
C) Round Robin
D) Priority

- ✓ **Answer:** C
- (MCQs 6–25 include background queues, scheduling among queues, advantages & disadvantages)

• ☒ **Short Questions (With Answers)**

- **Q1. What is multilevel queue scheduling?**
Answer: Scheduling where ready queue is divided into multiple fixed queues.
- **Q2. Can processes move between queues?**
Answer: No, processes are permanently assigned.
- **Q3. Difference between foreground and background queues?**
Answer: Foreground handles interactive jobs, background handles batch jobs.

• ☐ **Long Question (With Answer)**

- **Q. Explain Multilevel Queue Scheduling with diagram.**
- **Answer:**
Multilevel queue scheduling divides processes into multiple queues based on type. Each queue has its own scheduling algorithm and priority. It is simple but inflexible.

• ? **CS311 – LECTURE 17**

• **Multilevel Feedback Queue Scheduling (MLFQ)**

• ✓ **MCQs (25) – With Explanation**

- **Q1. Multilevel Feedback Queue Scheduling allows processes to:**
A) Stay in one queue
B) Move between queues
C) Run once
D) Execute in FCFS order
- ✓ **Answer:** B
Explanation: Processes can move between queues based on behavior and CPU usage.
- **Q2. MLFQ is designed to:**
A) Reduce memory
B) Improve response time
C) Eliminate CPU
D) Replace Round Robin
- ✓ **Answer:** B
- **Q3. Which scheduling feature makes MLFQ flexible?**
A) Fixed queues
B) Priority only
C) Feedback mechanism
D) FCFS policy
- ✓ **Answer:** C

- **Q4.** New processes usually enter:
 - A) Lowest priority queue
 - B) Middle queue
 - C) Highest priority queue
 - D) Random queue
- ✓ **Answer: C**
- ---
- **Q5.** CPU-bound processes in MLFQ are moved:
 - A) Upward
 - B) Downward
 - C) Removed
 - D) Blocked
- ✓ **Answer: B**
- ---
- **Q6.** I/O-bound processes are favored because:
 - A) They use less memory
 - B) They give up CPU quickly
 - C) They have high priority forever
 - D) They terminate early
- ✓ **Answer: B**
- ---
- **Q7.** Main objective of MLFQ is to:
 - A) Reduce starvation
 - B) Improve throughput
 - C) Optimize turnaround
 - D) All of these
- ✓ **Answer: D**
- ---
- **Q8.** MLFQ can suffer from starvation if:
 - A) No feedback
 - B) No aging
 - C) Only one queue
 - D) CPU idle
- ✓ **Answer: B**
- ---
- **Q9.** Aging in MLFQ helps to:
 - A) Reduce waiting
 - B) Increase priority
 - C) Avoid starvation
 - D) All of these
- ✓ **Answer: D**
- ---
- **Q10.** MLFQ scheduling is:
 - A) Static
 - B) Non-preemptive
 - C) Preemptive
 - D) Manual

- ✓ **Answer:** C
- (MCQs 11–25 cover queue levels, time quantum variation, advantages, disadvantages, comparison)

- ☒ **Short Questions (With Answers)**

- **Q1. What is Multilevel Feedback Queue Scheduling?**

Answer: A scheduling algorithm where processes can move between multiple queues based on execution behavior.

- **Q2. Why is MLFQ called “feedback” scheduling?**

Answer: Because scheduling decisions depend on past CPU usage of processes.

- **Q3. How does MLFQ prevent starvation?**

Answer: By using aging to increase priority of long-waiting processes.

- ☐ **Long Question (With Answer)**

- **Q. Explain Multilevel Feedback Queue Scheduling in detail.**

- **Answer:**

MLFQ scheduling divides ready queue into multiple levels. Processes can move between queues based on CPU usage. CPU-bound jobs move to lower priority queues, while I/O-bound jobs stay in higher priority queues. It improves response time and fairness but is complex to implement.

- ☐ **CS311 – LECTURE 18**

- **Process Synchronization**

- ✓ **MCQs (25) – With Explanation**

- **Q1. Process synchronization deals with:**

- A) CPU speed
- B) Memory allocation
- C) Shared resources
- D) Disk scheduling

- ✓ **Answer:** C

- **Q2. Critical section is:**

- A) Code accessing shared data
- B) Infinite loop
- C) Dead code
- D) System call

- ✓ **Answer:** A

- **Q3. Race condition occurs when:**

- A) CPU idle
- B) Processes run sequentially

- C) Shared data accessed concurrently
- D) Process blocked

- ✓ **Answer: C**

- **Q4.** Mutual exclusion ensures that:

- A) CPU always busy
- B) Only one process enters critical section
- C) All processes run together
- D) No deadlock

- ✓ **Answer: B**

- **Q5.** Which is NOT a requirement of critical section problem?

- A) Mutual exclusion
- B) Progress
- C) Bounded waiting
- D) Deadlock

- ✓ **Answer: D**

- **Q6.** Semaphore is used for:

- A) Memory allocation
- B) Process synchronization
- C) File handling
- D) CPU scheduling

- ✓ **Answer: B**

- **Q7.** Binary semaphore values are:

- A) 0 and 2
- B) 1 and 2
- C) 0 and 1
- D) -1 and 1

- ✓ **Answer: C**

- **Q8.** wait() operation on semaphore:

- A) Increments value
- B) Decrements value
- C) Deletes semaphore
- D) Unlocks CPU

- ✓ **Answer: B**

- **Q9.** signal() operation does what?

- A) Decrements semaphore
- B) Blocks process
- C) Increments semaphore
- D) Deletes semaphore

- ✓ **Answer: C**

- **Q10.** Busy waiting occurs in:

- A) Mutex

- B) Spinlock
- C) Monitor
- D) Scheduler

- ✓ **Answer: B**
- (MCQs 11–25 include Peterson's solution, semaphores, mutex, problems)

• ☒ **Short Questions (With Answers)**

- **Q1. What is race condition?**

Answer: A situation where multiple processes access shared data simultaneously causing inconsistent results.

- **Q2. What is a semaphore?**

Answer: A synchronization tool used to control access to shared resources.

- **Q3. Difference between mutex and semaphore?**

Answer: Mutex allows one process, semaphore allows multiple based on count.

• ☐ **Long Question (With Answer)**

- **Q. Explain process synchronization and critical section problem.**

- **Answer:**

Process synchronization ensures correct execution of concurrent processes accessing shared resources. Critical section problem ensures mutual exclusion, progress, and bounded waiting to avoid race conditions.

• ☐ **CS311 – LECTURE 19**

• **Deadlock**

• ✓ **MCQs (25) – With Explanation**

- **Q1. Deadlock occurs when:**

- A) CPU idle
- B) Processes wait indefinitely
- C) Memory full
- D) Disk failure

- ✓ **Answer: B**

- **Q2. Which condition is NOT necessary for deadlock?**

- A) Mutual exclusion
- B) Hold and wait
- C) Preemption
- D) Circular wait

- ✓ **Answer: C**

- **Q3. Deadlock prevention means:**

- A) Ignoring deadlock
- B) Eliminating one condition

- C) Detecting deadlock
- D) Recovering deadlock

- ✓ **Answer: B**

- **Q4.** Deadlock avoidance uses:

- A) Semaphore
- B) Banker's Algorithm
- C) Round Robin
- D) FCFS

- ✓ **Answer: B**

- **Q5.** Safe state means:

- A) No deadlock
- B) Deadlock occurred
- C) CPU idle
- D) All blocked

- ✓ **Answer: A**

- **Q6.** Resource allocation graph is used for:

- A) Detection
- B) Prevention
- C) Avoidance
- D) Scheduling

- ✓ **Answer: A**

- **Q7.** Deadlock detection requires:

- A) Preemption
- B) Resource graph
- C) Priority
- D) Time quantum

- ✓ **Answer: B**

- **Q8.** Starvation differs from deadlock because:

- A) Starvation is temporary
- B) Deadlock is permanent
- C) Starvation may resolve
- D) All of these

- ✓ **Answer: D**

- **Q9.** Recovery from deadlock can be done by:

- A) Process termination
- B) Resource preemption
- C) Rollback
- D) All of these

- ✓ **Answer: D**

- **Q10.** Banker's Algorithm checks system for:

- A) Unsafe state

- B) Safe state
- C) Deadlock only
- D) Priority

- ✓ **Answer: B**
 - *(MCQs 11–25 include deadlock handling strategies, examples, comparisons)*
-

- ☒ **Short Questions (With Answers)**

- **Q1. What is deadlock?**
Answer: A condition where processes wait forever for resources held by each other.
 - **Q2. List four deadlock conditions.**
Answer: Mutual exclusion, hold and wait, no preemption, circular wait.
 - **Q3. What is deadlock avoidance?**
Answer: Technique to avoid deadlock by ensuring system stays in safe state.
-

- ☐ **Long Question (With Answer)**

- **Q. Explain deadlock and methods to handle it.**
- **Answer:**
Deadlock occurs when processes are permanently blocked. Deadlock handling methods include prevention, avoidance (Banker's algorithm), detection, and recovery. Each method has advantages and trade-offs.

WU
Updates Tech