CS Exit Ethiopia

Revision Outlines

Computer Science Exit Examination

1 selected course | 18 November 2026

Selected courses

01. Web Programming

https://csexitethiopia.com

abelasnake2003@gmail.com | Telegram: @Abechu7

Course contents

Each course begins on a new page. PDF bookmarks provide question, study-note section and revision-topic navigation.

Web Programming 3

Web Programming

Revision Outlines

Browser, server and technology roles | Exam focus

- Internet, Web and service roles
  - Internet: connected networks
  - Web: resource system, commonly HTTP
  - Browser/user agent: requests and HTML/CSS rendering
  - Permitted JavaScript execution
  - HTTP server: requests→representations
  - Server may invoke application code
  - Client/server roles can share one machine
- Technology responsibilities
  - HTML: structure/meaning; CSS: presentation
  - JavaScript: programmable interactivity
  - PHP; Python; JSP/Java; ASP.NET
  - Server responses and database access
  - C#: commonly ASP.NET
  - Node.js: JavaScript runtime, not language
  - JavaScript on client or server
  - Browser sandbox; permission-based file/device APIs
  - Nontrivial programs possible
  - No unrestricted arbitrary client-hardware access

Hosting, URLs and static versus dynamic content | Exam focus

- Hosting, addressing and URLs
  - Hosting: serving environment
  - Domain registration: reserved name
  - DNS: names→network records
  - ISP: connectivity
  - URL: scheme, host, optional port
  - Path; query; fragment

- Host: machine/service name
- Fragment: local document location

• Static versus dynamic content

- Static server: stored files
- Dynamic: logic/request/data-built representations
- Static pages may run interactive scripts
- Dynamic content need not visibly change
- Website: resource group; page: document/resource
- Open standards ≠ identical browser support
- FrontPage; Macromedia HomeSite5: historical editors
- Editors ≠ languages or modern requirements

Read an HTTP/1.1 exchange | Exam focus

• HTTP/1.1 message anatomy

- Request: method, request-target, version
- Headers→blank line→optional body
- GET /courses?id=7 HTTP/1.1
- Host: example.edu; terminating blank line
- Body may be absent
- Response: version, status, optional reason
- Headers→blank line→permitted content
- Request line ≠ response status line

• HTTP representation, framing and state

- Content-Type: representation format
- Content-Length/transfer framing: message boundaries
- Stateless request semantics
- Sessions/databases maintain application state
- HTTP2/3 encode differently
- Request-line universality requires HTTP/1.x scope
- Asynchronous client use still receives responses

Methods, status and GET/POST reasoning | Exam focus

• HTTP methods and intended effects

- GET: safe representation retrieval
- Intentional state changes: avoid GET links
- POST: process supplied content
- PUT: create/replace identified representation
- DELETE: remove resource association
- Safe: intended state effects
- Idempotent: repeated same intended effect
- Same response not required
- GET/PUT/DELETE idempotent; arbitrary POST not

• Status codes and form transport
- 200 success; 201 creation; 204 no-content
- 301 permanent redirect; 400 malformed request
- 401 missing/invalid authentication; 403 refusal
- 404 unavailable/not found; 500 unexpected failure
- GET forms: URL query name/value pairs
- POST forms: body name/value pairs
- Neither method encrypts
- POST body visible in developer tools
- Server/configuration imposes size limits

HTTPS and resource-oriented APIs | Exam focus

• HTTPS and REST constraints
- HTTP over TLS
- Channel confidentiality/integrity; certificate-based server authentication
- No input-safety or merchant-trust guarantee
- REST: client/server separation
- Stateless requests; cacheability; uniform resource interface
- Resources: identifiers and representations
- JSON common, not required
- JSON output alone ≠ REST

• Resource API reasoning
- /students/7: GET read, PUT replace
- DELETE requests removal

- Authentication; authorization; validation
- Consistent error representations
- Retry safety before repeating requests
- Durable facts stored server-side
- Stateless requests can use persistent databases

HTML document, text and metadata | Exam focus

- HTML document structure and metadata
  - HyperText Markup Language
  - <!DOCTYPE html>: modern standards mode
  - html root; head metadata; body content
  - title: page/tab title
  - Meta charset: encoding
  - Description: summary; viewport: mobile viewport behavior
  - h1-h6: heading hierarchy
  - h6 conventionally smallest default
  - CSS size not mandated
- Text semantics and element anatomy
  - p: paragraph; br: line break
  - hr: thematic break
  - pre: preserved whitespace; code: code
  - strong: importance; em: stress
  - b/i semantics ≠ importance/emphasis
  - sub/sup: subscript/superscript
  - <!-- comment -->: hidden display, visible source
  - Entities: © < > &
  - Tags: boundaries; element: content/attributes
  - Void element example: img

Links, images and accessible meaning | Exam focus

- Links and browsing targets
  - Read
  - href: destination, not url/link

- href="#details"→id="details"
- _self: current; _blank: new context
- _parent: parent; _top: top-level
- Browser preference: tab versus window

• Images and accessibility
- img src: source; alt: alternative
- Informative images: meaningful alternative text
- Decorative images: empty alt
- Width/height reserve dimensions
- title ≠ replacement for alt
- Anchor may wrap image
- Programmatic labels; clear links; keyboard access
- Visual styling alone ≠ accessibility
- Attributes in opening tags
- Unique document-identifying IDs

Lists, tables and navigation | Exam focus

• Lists and navigation
- ul: unordered; ol: ordered
- li: list items
- Ordered list: start and numbering type
- CSS list-style-type: presentation
- dl: definition list
- dt: term; dd: description
- Semantic nav: list of links

• Tables and navigation reset
- table→tr rows→th/td cells
- th: header; td: data, not table
- colspan: columns; rowspan: rows
- Account for previously occupied cells
- Accessible header associations
- CSS borders, background, padding, border-collapse
- Legacy bgcolor/bordercolor not preferred

- nav ul: list-style-type:none;margin:0;padding:0
- Reset removes bullets/default spacing only
- Reset ≠ horizontal links/dropdown

Form controls and successful submission | Exam focus

• Form submission and controls
- action: processing URL
- method: submission; enctype: encoding
- Default method: GET
- Named enabled successful controls contribute values
- Unchecked checkbox/radio usually absent
- label for→control id
- name: submitted field identifier
- Same-name radios: single choice
- Checkboxes: independent selections
- select→option; textarea: multiline

• Input behavior and validation limits
- size: approximate visible text width
- maxlength: entered-text length limit
- Neither establishes server security bound
- Password input masks, never encrypts
- submit: submission; button: no intrinsic submission
- email/required: browser validation feedback
- Arbitrary clients bypass validation
- GET/POST ≠ universal per-element parsing

Frames, media and graphics

• Legacy frames and iframe
- frameset/frame: separate window documents
- Old frameset replaces body
- Named targets select changed frame
- Independent scrolling leaves other frames stationary
- Frameset/frame obsolete in modern HTML

- iframe embeds document within ordinary page
- Untrusted iframe: appropriate permissions/sandboxing

• Media and graphics
- audio/video: supported media, controls, fallback
- embed/object may embed supported resources
- Generic video-tag question may be nonexclusive
- bgsound/Flash history ≠ current support
- Canvas: scripted bitmap drawing context
- SVG: scalable DOM shapes
- Canvas getContext("2d")→fillRect(x,y,w,h)
- Paths: beginPath/moveTo/lineTo/stroke
- Informative graphics require text alternatives

CSS attachment and selector meaning

• CSS attachment and syntax
- External: link rel="stylesheet" href="styles.css"
- External link in head
- Internal: style; inline: style attribute
- External reuse, consistency, caching
- No universal smaller-page/accessibility guarantee
- selector { property: value; }
- CSS comments: /* ... */

• Selector meanings
- p: type; .notice: class; #logo: ID
- h1,h2: grouped selectors
- nav ul: descendants; nav>ul: direct children
- [type="email"]: attribute match
- :hover/:focus: pseudo-class states
- ::before: pseudo-element part
- Selector chooses declaration targets
- Selector ≠ color-setting property

Resolve cascade, specificity and inheritance | Exam focus

- Cascade winner procedure
  - Origin, importance, layer context first
  - Then specificity and source order
  - Same-origin/layer author context assumed
  - ID>class/attribute/pseudo-class>type/pseudo-element
  - Counts compared lexicographically
  - Not unlimited base-ten total
  - Equal specificity: later applicable rule
  - Important beats normal within relevant context
- Specificity, importance and inheritance
  - p#intro.note example
  - #intro blue beats .note red
  - p green!important beats both normal rules
  - Inherited color/fonts: no stronger specified value
  - Margin/padding/border normally not inherited
  - :not(): argument specificity
  - Module universal-zero simplification incorrect

Style properties and box calculations | Exam focus

- CSS properties and box parts
  - color: text; background-color: background
  - font-size: size; font-family: face
  - font-style: italic; font-weight: weight
  - text-align: inline-content alignment
  - text-decoration: underline effects
  - Margin outside border
  - Padding inside border
  - Border surrounds content/padding
- Box-model calculations and relative units
  - Content-box: width200,padding10,border2→224px
  - Border-box width200→content176px
  - Margins excluded from border-box width

- Shorthand: top/right/bottom/left
- One: all; two: vertical/horizontal
- Three: top/horizontal/bottom; four: each side
- Adjacent vertical block margins may collapse
- Padding never collapses
- Non-replaced inline width/height differs from blocks
- rem: root font size
- font-size em: parent font size
- Many other em lengths: element computed-font-size

• Historical body color attributes
- text: default text color
- bgcolor: background
- Modern CSS: color and background-color

Layout, alignment and responsive design | Exam focus

• Flow, flex, grid and alignment
- Normal flow: blocks/inlines
- Flexbox: direct children, main axis
- justify-content: main; align-items: cross
- Horizontal row: vertical cross-axis alignment
- Grid: explicit rows and columns
- grid-template-columns:1fr 2fr
- vertical-align: inline/table-cell contexts
- vertical-align ≠ universal block centering
- align-content: extra-space track/line distribution
- align-content ≠ arbitrary single-child centering

• Positioning and responsive design
- Relative: offset, retained flow space
- Absolute: removed from flow, containing block
- Positioned ancestor often establishes containing block
- Fixed: commonly viewport
- Flexible layout/images; viewport metadata; media queries
- @media(max-width:600px){.cards{grid-template-columns:1fr}}

- min-width768px: wider viewports, not mobile-only
- Compatibility: feature tests and fallbacks
- Compatibility ≠ responsive size handling

JavaScript syntax, inclusion and declarations

- JavaScript inclusion and language identity
  - External .js: <script src="app.js"></script>
  - JavaScript source: no HTML closing-tag syntax
  - defer: after parsing, before DOMContentLoaded
  - Originated at Netscape; distinct from Java
  - Prototype-based objects; first-class functions
  - Sophisticated applications supported
- JavaScript lexical and declaration rules
  - Case-sensitive identifiers
  - Initial letter/underscore/$; no digits/spaces
  - var: function/global scope
  - let/const: block scope
  - const: no rebinding, object mutation allowed
  - // line comments; /* */ blocks
  - Semicolons and automatic insertion
  - Newlines can affect parsing after return
  - "use strict": directive string, not keyword
  - Parse syntax before output computation

Types, conversions and missing values

- JavaScript value categories
  - Primitives: undefined,null,Boolean,Number,BigInt,String,Symbol
  - Objects: arrays, functions, dates
  - Float not separate Number primitive
  - Arrays technically objects
  - typeof new Date()→"object"
  - typeof null→"object": historical behavior
- Missing values and conversion

- Declared unassigned/missing ordinary property→undefined
- Undeclared direct read→usually ReferenceError
- Either string quote style; appropriate escapes
- String-involved + may concatenate
- Arithmetic / converts numeric strings
- "abc"/"def"→NaN
- Nonzero Number/0→signed Infinity; 0/0→NaN
- parseInt("12.8",10)→12, stops at dot
- Parsing ≠ rounding
- Global isNaN converts first
- Number.isNaN does not coerce

Trace assignments, comparisons and postfix updates | Exam focus

• Operator tracing
  - Record value and type
  - Specified evaluation order
  - ==/!=: coercion
  - ===/!==: no coercion
  - Strict operators coexist with loose operators
  - condition?x:y selects one branch
• Assignment, concatenation and postfix microtraces
  - var a="20";var b=a=30
  - Both Number30; a+b=60
  - Supplied key50 incorrect
  - "1 plus 1 is "+1+1
  - Left-to-right result: "1 plus 1 is 11"
  - Postfix a++ yields old value
  - Increment follows yielded value
  - a=10;a==a++→10==10;body sees11
  - a=1;print a-- then a→10
  - JS sequencing not portable to C/C++
  - C/C++ unsequenced modifications trap

Math, string and array API microtraces

- Math and substring traces
  - Math.abs(-9)=9
  - Math.floor(5.9)=5
  - Math.sqrt(36)=6
  - Zero-based indexes
  - substring(start,end): start inclusive, end exclusive
  - "exitexam".substring(2,6)→"itex"
- Array operations and reduction
  - push/pop: end; unshift/shift: beginning
  - slice: selected shallow copy
  - splice(start,deleteCount,...items): mutating removal/insertion
  - reverse: in-place reversal
  - reduce: sequential left-to-right accumulator
  - reduceRight: sequential right-to-left, not parallel
  - [1,2,3].reduceRight((a,v)=>a-v,0)
  - Accumulators: −3,−5,−6
  - No initial value: last present item
  - Empty without initial value: throws
  - Exact method spelling and case

Control flow, functions and errors | Exam focus

- Loop execution order
  - for: initialize once→test→body→update
  - while: pretest
  - do...while: at least one execution
  - break exits; continue advances iteration
  - n=1;n<7;n++ with n%5==0
  - Message only at n=5
- Functions and error handling
  - function f(x){return x*2;}
  - Parameter; return value; stored/passed functions
  - Arrow: compact, different this/constructor behavior

- Runtime argument limits reject absolute unlimited claim
- try/catch: thrown runtime errors
- throw: signal; finally: cleanup
- Syntax versus runtime versus logical errors
- bool ans=true invalid; let ans=true valid
- Literal "type of" ≠ typeof operator

DOM selection, creation and safe updates | Exam focus

- DOM selection and timing
  - DOM: hierarchical document nodes
  - getElementById("x"): element or null
  - querySelector(".item"): first match
  - querySelectorAll: static NodeList
  - Other plural lookups may be live
  - Guard missing elements
  - Run after relevant markup exists
- DOM creation, attachment and safe mutation
  - createElement("div")→textContent="Hello"→appendChild(el)
  - Creation alone does not insert
  - createTextNode: text; appendChild: attachment
  - innerHTML parses/replaces descendants
  - innerHTML+= reparses/replaces previous descendants
  - Listener/state destruction possible
  - Untrusted plain text: textContent
  - style.backgroundColor: style property
  - classList.add: class mutation
  - getElementByld: lowercase-l misspelling, nonexistent method

Events and validated submission | Exam focus

- Events and listener options
  - Events: click,input,keydown,mouseover,submit,load
  - addEventListener("click",handler): register, not call
  - Multiple listeners coexist

- once: remove after use
- capture: earlier propagation phase
- passive: promise no default cancellation
- Event bubbling ≠ browser default action

• Submission validation and cancellation
- submit listener checks valid()
- Failure only: e.preventDefault()
- addEventListener return false does not cancel
- Inline/onsubmit return false: caller-dependent cancellation
- Cancelable keyboard default: preventDefault
- Different key event need not disappear
- Browser checks: feedback
- Repeat validation/authorization server-side

Window, dialogs and browser properties

• Window, dialogs and browser properties
- window: browsing context; document: document property
- alert: information; confirm: Boolean
- prompt: text or cancellation null
- console.log: diagnostics
- location.reload(): current-page reload
- screen.availHeight; screen.availWidth; screen.colorDepth
- Exact camelCase; ColorsDepth invalid

• Globals, load and document hazards
- window.onload: load-handler property
- Load typically waits dependent resources
- Global let/const not window properties
- Classic top-level var bindings differ
- document.write: method/function property
- Post-load write may replace document
- document.bgColor: legacy property
- Document.bgcolor: wrong object/case
- Modern: document.body.style.backgroundColor

- Workers: concurrent computation
- DOM normally page main thread

PHP syntax, names and types

- PHP syntax and legacy delimiters
  - PHP: Hypertext Preprocessor
  - .php file; <?php ... ?>
  - Final closing tag optional
  - Emits ordinary text and HTML
  - PHP8 rejects <script language="PHP"> delimiters
  - Short <? configuration-dependent
  - Prefer full opening form
  - Stray >> not closing tag
- PHP variables, types and constants
  - $ then letter/underscore
  - Subsequent letters/digits/underscores
  - $AGE18 valid; $18AGE invalid
  - Case-sensitive variable names
  - Dynamic typing; optional type constraints
  - Scalars: int,float,string,bool
  - Structured: arrays/objects; null: no value
  - var_dump: type and value
  - Constants normally not redefined during execution

PHP output, strings and comparisons

- PHP output and concatenation
  - echo: language construct, no return
  - echo: comma-separated expressions
  - print: one expression, returns1
  - printf/print_r: other output mechanisms
  - Neither inserts spaces/newlines automatically
  - print5 twice→55
  - "I"."am"→Iam

- Dot concatenates; + arithmetic

- PHP string and comparison traces
  - Double quotes: variables and escapes
  - Single quotes: dollar names normally literal
  - Literal $lfc: single quotes/escaped dollar
  - Zero-based offset: "maroon"[2]→r
  - Surrounding PHP must be valid
  - strlen: bytes
  - strpos: integer0 or false
  - Failure test: ===false
  - strrev: reverse bytes; str_replace: replace matches
  - 8==8.0 true; 8===8.0 false
  - echo false→empty; echo true→1
- Raw versus rendered PHP output
  - PHP sends markup characters
  - Browser interprets response
  - echo "A
B": raw tag, displayed lines
  - Malformed </br>: HTML error recovery
  - Specify raw versus rendered trace

PHP scope, parameters and persistent locals | Exam focus

- PHP scope and static storage
  - Outside variable: global scope
  - Function locals/by-value parameters: separate bindings
  - Explicit global $x or $GLOBALS["x"]
  - Superglobals available across scopes
  - Static local persists across lifecycle calls
  - Static storage ≠ merely visibility scope
  - Uninitialized static local: null
  - Echo uninitialized static: no characters
- Parameter and static-counter traces
  - Global $i=10; sample($i) by value
  - Function increments/prints local parameter

- Caller remains10
- Before,during,after→10,11,10
- Commas only when source writes them
- External global declaration ≠ reference parameter
- Teaching taxonomy: local,global,static
- Static counter0; successive increments→1,2

PHP arrays, loops and callable behavior | Exam focus

- PHP array keys and syntax
  - Integer/string keys map to values
  - Unspecified numeric indexes start0
  - [20,12,23,19,18]: index3→19
  - ["top"=>"black"]: top→black
  - Contiguous => required
  - Split = >: parse error
- PHP loops and dynamic behavior
  - foreach($items as $key=>$value): entries
  - for/while: tests and updates
  - $sub-=$sub means $sub=$sub-$sub
  - Self-subtraction ≠ decrement
  - $sum=0;foreach([2,4]) accumulation→6
  - Functions: arguments, return, output
  - Classes/objects/variables also support server processing
  - No single mechanism defines dynamic PHP
  - Explicit braces clarify nesting

Request data, server validation and injection | Exam focus

- PHP request sources and existence
  - $_GET: query; $_POST: supported form POST
  - $_FILES: upload metadata
  - $_SERVER: request/environment
  - $_COOKIE: received cookies
  - $_SESSION: session data after startup

- $GLOBALS: global-symbol array
- $_GLOBAL: not built-in
- $_REQUEST: configured GET/POST/cookie combination
- Ambiguous precedence; prefer intended source
- $_REQUEST ≠ database interface
- isset: exists and non-null
- isset ≠ truth or valid syntax

• Secure request-processing sequence
- Read→required/type/length/range checks
- Authenticate→authorize→process
- SQL binding separates values from commands
- Contextual HTML/attribute/JS output encoding
- textContent for browser plain text
- Injection/XSS testing detects, never itself fixes
- Client validation/encryption/connection cannot replace controls

Cookies and browser storage | Exam focus

• Cookies and PHP headers
- Browser-held name/value data
- Automatic attachment only to matching requests
- Domain; path; expiry; Secure; SameSite
- HttpOnly: restricted script access
- Secure: secure transport required
- Cookies not indiscriminately sent everywhere
- setcookie sends response header
- Before output unless buffering defers output
- $_COOKIE: received values

• Browser storage and session interpretation
- localStorage: origin-specific strings, persistent until cleared
- sessionStorage: page session/tab scope
- Neither automatically transmitted like cookies
- Scripts can explicitly send stored values
- IndexedDB: structured local storage

- Cookies/localStorage/sessionStorage/IndexedDB: client storage
- localStorage tokens: script-access risk
- Storage limits vary
- Reopen login: persistent/remember token or restoration
- Stored data alone ≠ server acceptance

Server sessions and the authorization trust path | Exam focus

- PHP server-session lifecycle
  - session_start(): start/resume, enable $_SESSION
  - Browser: opaque session identifier
  - Session data: server-controlled storage
  - Missing views→1; later requests increment
  - unset: one entry
  - Destroyed session data ≠ cleared browser cookie
  - Current variables separately cleared
  - Logout handles complete lifecycle
- Session authorization trust path
  - Client logged_in=1 cookie: attacker-controlled
  - Flag-only trust permits authorization bypass
  - Valid session ID plus server user/role
  - Check every protected action
  - Unrelated flag edit grants no rights
  - HTTPS ≠ trustworthy client claims
  - Omitted server check: scenario indeterminate
  - Protect IDs; regenerate after login
  - Appropriate expiration; no sensitive URL tokens

Uploads, files and storage safety | Exam focus

- Secure upload workflow
  - POST; multipart/form-data; named file input
  - accept: browser hint, not server enforcement
  - Validate server-side size/content
  - Inspect upload errors

- Use temporary upload path
- Safe generated destination; move_uploaded_file
- Avoid executable storage
- Distrust client filenames/extensions

• File APIs and mode traces
- file_exists: file or directory
- is_file: regular file
- fopen: stream or false
- r: existing read; r+: read/write, untruncated
- w: truncate/create write; w+: also read
- a: append/create; x: exclusive creation
- fclose: release stream
- readfile: emit content, return byte count
- echo readfile(...): also prints count
- Check failures before access
- Constrain paths to permitted locations

Database connections, TLS and actual APIs | Exam focus

• PHP database connections
- MySQL: relational DBMS
- Persistent structured data; concurrent clients
- PHP8: MySQLi or PDO
- Removed mysql_* extension requires migration
- mysqli_connect(host,user,password,database,port,socket)
- Database selection may be deferred
- Defaults make multiple arguments technically optional
- Exactly-one-optional-field claim incorrect

• Database TLS and API signatures
- MySQLi: mysqli_init()→mysqli_ssl_set(...)→mysqli_real_connect(...)
- Required verification configuration
- Nonstandard: mysqli_ssl_connect(); pdo_connect_ssl()
- Nonstandard: mysql_ssl_connect(); secure_connect()
- PDO: driver DSN/options

- Procedural MySQLi, object methods, PDO differ
- affected_rows() alone not generic PHP function

SQL operations, result rows and safe rendering | Exam focus

- SQL operations and identifier generation
  - SELECT retrieves; INSERT adds
  - UPDATE ... SET modifies; DELETE removes
  - WHERE limits matching/affected rows
  - DELETE without WHERE: ordinarily all rows
  - Constraints, permissions, errors still apply
  - CREATE DATABASE/TABLE: storage definitions
  - Primary key identifies rows
  - AUTO_INCREMENT: numeric values, gaps/explicit values possible
  - Not gapless business numbering
- MySQLi results and safe rendering
  - mysqli_query: result object for result queries
  - Other successful statements: true
  - Failure/exception depends on settings
  - mysqli_fetch_assoc: associative rows
  - mysqli_fetch_row: positional; mysqli_fetch_object: objects
  - mysqli_num_rows: result-row count
  - mysqli_affected_rows($conn): changed/inserted/deleted, options-dependent
  - mysqli_insert_id($conn): generated AUTO_INCREMENT value
  - Escape rendered cells; prepare submitted values
  - mysqli_close: connection close

PHP includes, objects and dates | Exam focus

- PHP includes and version-dependent failure
  - include: current-scope evaluation; failure warning
  - require: critical failure
  - PHP8 failed require raises catchable Error
  - Uncaught Error stops normal execution
  - E_COMPILE_ERROR description: older PHP

- include_once/require_once: no repeated inclusion
- No arbitrary user-selected include path

• Objects and explicit time assumptions
- Class: properties/methods; new: instance
- $obj->field: property; $obj->method(): call
- public: external; private: class-restricted
- Constructor initializes instance
- $this: instance-method context
- DateTime/DateTimeZone: time and zone
- diff→DateInterval; DatePeriod iterates intervals
- Explicit zones and calendar assumptions
- Do not assume server-local timezone

AJAX and response handling | Exam focus

• AJAX and XHR workflow
- Asynchronous JavaScript and XML
- Responses: JSON,text,HTML,XML
- Technique, not language/standalone framework
- XML not mandatory
- open("GET",url,true)→register handlers→send()
- Response and error handlers
- onreadystatechange may fire repeatedly
- readyState4: completed
- Then check status and data

• Callbacks, fetch and response ordering
- Callback: runtime/API-invoked later function
- Browser invokes available-response handler
- Remote server never directly executes browser code
- await fetch(url)→response
- Check response.ok; await response.json()
- Handle network and parse errors
- Asynchronous result unavailable to immediate next statement
- Older late response must not overwrite newer

Polling, WebSockets and data formats | Exam focus

- Polling versus WebSocket
  - Short polling: repeated update requests
  - Long polling: hold request until event/timeout
  - Respond, finish, then client reconnects
  - Long polling ≠ same-response continuous unsolicited stream
  - WebSocket: handshake then persistent bidirectional messages
  - AJAX/WebSocket both allow nonblocking interaction
  - Both carry JSON; broad stems nonexclusive
- Formats and transport decisions
  - JSON: objects,arrays,strings,numbers,Booleans,null
  - Double-quoted keys/strings
  - Functions/undefined not JSON values
  - XML: valid interchange alternative
  - REST can return JSON or XML
  - Select latency, direction, connection cost, recovery
  - Asynchronous alone insufficient transport criterion
  - Validate messages; authorize each operation

Node server, event loop and scaling | Exam focus

- Minimal CommonJS Node HTTP server
  - const http=require("node:http")
  - http.createServer((req,res)=>{...}).listen(8000)
  - Incoming requests invoke listener
  - res.writeHead(200,{"Content-Type":"text/plain"})
  - res.end("Hello"): complete response
  - Define startup-log variables
  - Module hostname otherwise undefined
- Event-loop concurrency and scaling
  - Event loop schedules JavaScript callbacks
  - OS/worker facilities: asynchronous work
  - Concurrency: overlapping progress
  - Parallelism: same-instant execution

- One loop thread handles waiting requests
- Long CPU loops/synchronous I/O block callbacks
- Workers/processes/load balancer distribute work
- Shared data/session coordination not automatic
- Multiple clients or overlapping single-client requests

Races, locks and transaction invariants | Exam focus

- Race conditions and synchronization
  - Two requests read stock1
  - Both accept sale and write0
  - Two sales consume one item
  - Interleaving-dependent correctness despite short synchronous callbacks
  - Mutex: critical-section protection
  - Semaphore: bounded access
  - Spinlock: busy-wait
  - Spinlocks unsuitable for browser/event-loop blocking
  - Several mechanisms synchronize; mutex not unique
- Database invariants and retry safety
  - Atomic conditional stock decrement
  - UPDATE stock SET qty=qty-1
  - WHERE id=? AND qty>0
  - Check affected rows
  - Related writes in appropriate transaction
  - Row lock/serializable transaction for multistep logic
  - Conflict rollback/retry
  - Process-local lock misses other balanced servers
  - Idempotency tokens plus unique storage constraints
  - Retried payment must not duplicate transaction

Frameworks and version-aware setup | Exam focus

- Frameworks, libraries and components
  - Framework: conventions/reusable structure
  - Library: callable capabilities

- jQuery: DOM simplification
- React: component interfaces
- Angular: component platform, distinct from AngularJS
- Vue: interactive interfaces
- Responsive-interface description not uniquely identifying
- React props: inputs; state: changing data
- JSX requires transformation
- JSX ≠ ordinary browser JavaScript syntax

• Version-aware framework setup
- Bootstrap: layout/components
- jQuery→Popper→Bootstrap: version4-era order
- Bootstrap5 drops required jQuery
- Angular CLI: ng new,ng serve,ng build
- ng generate; ng test
- e2e support configuration-dependent
- ng serve starts the development server
- Create React App: deprecated historical tooling
- Historical module tooling ≠ universal current recommendation

Manage content, performance and revisions | Exam focus

• Content, persistence, history and deployment
- CMS: editing, publishing, roles, media
- No per-file source edits required
- Database: persistent facts
- Database ≠ entire server application
- VCS: history, branches, merges
- Review and collaboration
- Deployment: publish chosen build
- Deployment separate from version control

• SEO and performance mechanisms
- SEO: meaningful content, headings, metadata
- Accessible links; crawlable structure
- Keyword metadata alone ≠ ranking guarantee

- Caching: reused representation/result, validity/invalidation rules
- Server cache: rendered pages/query results
- Unsafe shared key can expose private content
- Minification: remove source characters
- Bundling: combine; compression: transfer encoding
- Preloading: anticipated early fetch
- Excess bundling/preloading may compete with critical-work
- Measure performance
- Responsive and cross-browser testing separately

Assemble and evaluate an end-to-end feature | Exam focus

• Course-search browser feature
- Semantic headings/navigation; labeled search form
- Responsive grid; single-column fallback
- Submit registration after parsing
- Prevent default for asynchronous implementation only
- Encode query values
- GET /courses?q=...
- Loading state and useful errors
- No untrusted HTML insertion

• Search/enrollment end-to-end verification
- Validate query length/type; authenticate when required
- Parameterized SELECT; bounded results
- JSON with appropriate status/content type
- Client: status→parse→textContent nodes
- Enrollment: POST, CSRF, server authorization
- Capacity checks; transaction; success after commit
- Test empty/bad input and access denial
- Network failure; delayed responses; repeated clicks
- Simultaneous enrollment; narrow screens; keyboard use
- All four blueprint outcomes integrated

Validate syntax, context and output

- Literal source and trace preconditions
  - Valid syntax; known initial values
  - Execution context; raw/rendered representation
  - x=4+"4"5: missing-operator syntax error
  - bool ans=true: invalid declaration
  - type of new Date(): syntax error
  - PHP >> instead of ?>: parsing defect
  - Trace output only for valid syntax
  - Split = > invalid without authorized repair
- Answerability and output interpretation
  - Response text ≠ browser rendering
  - Emitted markup changes displayed output
  - Visual breaks may vanish in plain-text choices
  - Missing correct60 leaves no correct option
  - Storage/interface libraries/script languages/channels overlap
  - Broad descriptions need narrower distinction
  - Record the question number
  - Check assumptions before choosing
- Check exact code and punctuation
  - JavaScript coercion: x=4+"4"
  - Original output44, answer A
  - Extracted extra5: different invalid program
  - PHP closing delimiter: ?>
  - Original output r, answer D
  - Extracted >> falsely creates syntax error
  - Read the code exactly as shown
  - Trace strings and numeric conversions
  - Current extracted text: identify conflict