CS Exit Ethiopia

Complete Study Notes

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 6

How to use these study notes

START01 A practical study cycle

Start with a small set of questions before reading.

Mark the exact thing that stopped you: an unknown term, a rule, a calculation, a trace or a choice between alternatives.

Use the question lookup to open the relevant section.

Read one section, close it, and explain the rule in your own words.

Solve the question again and say why each other choice fails.

Then change one assumption or input and solve the variation.

This checks understanding beyond recognition.

A useful 60-minute session: 10 minutes of cold questions, 20 minutes of focused reading, 20 minutes of closed-book solving and 10 minutes updating an error log.

Adjust the timing to your needs.

Revisit mistakes after a gap, for example the next day, several days later and the following week.

These intervals are a practical starting plan, not a fixed learning law.

Record the section ID, your mistaken rule, the corrected rule and one boundary case.

Keep the log short enough to review.

Mix related courses during revision: programming with data structures; databases with security; networking with administration.

Explain how the same word changes meaning across contexts.

START02 What to emphasize

Emphasize definitions, key differences, algorithm costs, SDLC models, databases, networking and security.

Use short oral questions to check understanding and correct common misconceptions.

Combine those habits with the tracing, calculation, design and diagnosis demanded by the Ministry learning outcomes.

Allocate revision by both exam share and your error log.

A course with fewer items can still deserve more time if your foundations are weak.

The blueprint is a planning baseline, not a prediction of exact questions.

Definitions: distinguish nearby concepts, not just their names.

Calculations: write units, formulas, assumptions and intermediate steps.

Code and SQL: check validity before tracing the text; state any changes before analyzing a corrected version.

Algorithms and protocols: know the input, decision rule, state change, termination and failure cases.

Design choices: state the criterion.

A choice can change when cost, reliability, consistency or latency matters more.

Course group	Items in the 100-item blueprint	Revision implication
Web Programming	9	Reserve extra practice for HTML, CSS, JavaScript, PHP, client/server behavior and safe data handling.
Data Structures and Algorithms	7	Trace operations and compare their assumptions and costs.
Each of the other fourteen courses	6 each	Study all sixteen; do not abandon a whole course based on a short note or an alleged repeated-question list.

START03 How to handle exam traps

Underline NOT, FALSE, BEST and EXCEPT before comparing choices.

Check whether the stem supplies enough facts to select one answer.

A long scenario is not automatically an analysis question, and a familiar answer letter is not evidence.

When code is invalid, a diagram is absent, or two choices express the same rule, identify the clarification needed before answering.

Learn the correct rule and the smallest clarification needed.

Do not memorize an arbitrary key as a universal fact.

Distinguish worst case from average case, and an upper bound from a tight bound.

Check language version, SQL dialect, addressing assumptions, graph weights and tie-breaking rules.

For subnetting and scheduling, calculate rather than guess from a familiar number.

Use timed mixed papers only after you can explain your errors.

Revisit the reading sections behind recurring mistakes.

Web Programming

Complete Study Notes

WEB01 | Browser, server and technology roles

The Internet connects networks; the Web is a resource system commonly using HTTP.

A browser/user agent requests resources and renders HTML/CSS, then runs permitted JavaScript.

A web/HTTP server receives requests and returns representations, sometimes invoking application code.

Client and server are roles and can run on the same machine.

HTML supplies structure/meaning; CSS controls presentation; JavaScript supplies programmable interactivity.

PHP, Python, JSP/Java and ASP.NET applications generate server responses and access databases; C# is commonly used with ASP.NET.

Node.js is a JavaScript runtime, not a separate language.

JavaScript can run on either side.

Browser scripts are sandboxed, but may access selected files/devices through permission-based APIs.

They are not inherently limited to trivial programs, nor can they freely access arbitrary client hardware.

Web Programming module

WEB02 | Hosting, URLs and static versus dynamic content

Hosting supplies an environment where a site is served; domain registration reserves a name, and DNS maps names to network records.

An ISP provides connectivity.

A URL identifies scheme, host, optional port, path, query and fragment; the host identifies the machine/service name, while a fragment selects a local document location.

A basic static server returns stored files; dynamic processing constructs a representation from logic, requests or data.

Static pages can still run interactive browser scripts, and dynamic generation need not change visible text every time.

A website groups resources; a page is one document/resource.

Open standards do not guarantee identical support for every feature in every browser.

Historical editor-name items refer to Microsoft FrontPage and Macromedia HomeSite5, not languages or required modern tools.

Web Programming module

Adobe/Macromedia HomeSite support archive

Microsoft IE5 Corporate Deployment Guide

WEB03 | Read an HTTP/1.1 exchange

An HTTP/1.1 request begins with method, request-target and version, followed by header fields, a blank line, and an optional body.

For example, GET /courses?id=7 HTTP/1.1 followed by Host: example.edu and a terminating blank line can have no body.

A response begins with version, status code and optional reason phrase, then headers, blank line and permitted content.

Request lines are not response status lines.

Content-Type describes representation format; Content-Length or transfer framing establishes message boundaries when applicable.

HTTP is stateless in its request semantics, but applications maintain state through sessions/databases.

HTTP2/3 encode messages differently; a textual request-line description applies to HTTP/1.x.

Asynchronous use is a client programming choice, not proof that HTTP has no responses.

IETF RFC 9112

WEB04 | Methods, status and GET/POST reasoning

GET retrieves a representation with safe semantics; do not implement an intentional state-changing action as a GET link.

POST asks the target to process supplied content.

PUT creates/replaces the identified resource representation; DELETE requests removal of its association.

Safe describes intended state effects; idempotent means repeating the same request has the same intended effect, not necessarily the same response.

GET/PUT/DELETE are idempotent; arbitrary POST is not.

200 indicates success, 201 creation, 204 success without content, 301 permanent redirection, 400 malformed request, 401 missing/invalid authentication, 403 refusal, 404 unavailable/not found, and 500 an unexpected server failure.

GET forms usually put name/value pairs in the URL query; POST forms put them in the body.

Neither method encrypts data.

POST bodies are visible to developer tools and subject to server/configured size limits.

IETF RFC 9110

WEB05 | HTTPS and resource-oriented APIs

HTTPS uses HTTP over TLS, protecting channel confidentiality/integrity and authenticating the server under certificate checks.

It does not make unvalidated input safe or guarantee a trustworthy merchant.

A RESTful API follows constraints such as client/server separation, stateless requests, cacheability and a uniform resource interface.

Resources have identifiers and representations; JSON is common but not required.

Merely returning JSON does not prove REST.

For a resource /students/7, GET reads it, PUT can replace it, and DELETE requests removal, subject to authentication/authorization and validation.

Represent errors consistently and evaluate retry safety before repeating a request.

Store durable application facts server-side; stateless API requests can still operate on persistent database state.

Web Programming module

WEB06 | HTML document, text and metadata

HTML means HyperText Markup Language. <!DOCTYPE html> selects modern standards mode; <html> is the document root, <head> holds metadata, and <body> contains document content. <title> supplies the page/tab title.

Meta charset declares encoding; description summarizes the page; viewport controls mobile viewport behavior.

Heading levels h1-h6 express hierarchy, with h6 conventionally smallest in default styling, not a mandatory CSS size.

<p> marks paragraphs;
 is a line break and <hr> a thematic break. <pre> preserves source whitespace; <code> marks code. indicates importance and stress; <b>/<i> have other text semantics and are not identical to importance/emphasis. <sub>/<sup> mark sub/superscripts. <!-- comment --> is not displayed but remains visible in source.

Entities include ©, <, > and &.

Tags mark boundaries; an element includes content and attributes, or is a void element such as img.

Web Programming module

WEB07 | Links, images and accessible meaning

Use Read for a link; href is the destination, not url/link.

A fragment such as href="#details" links to an element with id="details". target="_self" uses the current context, _blank a new context, _parent its parent and _top the top-level context.

Browser preferences determine tab versus window.

<img src="photo.jpg" alt="Students in a lab"> supplies source and alternative text.

Use empty alt for purely decorative images; informative images need meaningful alternatives.

Width/height reserve dimensions; title is not a replacement for alt.

An anchor may wrap an image.

Give controls programmatic labels, clear link text and keyboard access; visual styling alone does not establish accessibility.

Attributes belong in opening tags, with unique IDs when used for document identification.

Web Programming module

WEB08 | Lists, tables and navigation

Unordered <ul> and ordered <ol> contain <li> items.

Ordered lists can specify start and numbering type; use CSS list-style-type for presentation.

A definition list uses <dl>, <dt> terms and <dd> descriptions.

Navigation can be a semantic <nav> containing a list of links.

<table> contains rows <tr>, header cells <th> and data cells <td>; a td is not the table itself. colspan spans columns; rowspan spans rows.

Account for occupied cells before placing the next cell, and use header associations for accessible data.

CSS handles borders, background, cell padding and collapsed borders; legacy bgcolor/bordercolor are not preferred styling.

Resetting nav ul with list-style-type:none; margin:0; padding:0 removes bullets and default spacing only; it does not itself arrange links horizontally or create a dropdown.

Web Programming module

WEB09 | Form controls and successful submission

A form’s action identifies its processing URL, method its submission method, and enctype its encoding.

The default method is GET.

Named enabled successful controls contribute values: unchecked checkboxes/radios are usually absent.

A label’s for matches the control’s id; name identifies its submitted field.

Same-name radio buttons form a single-choice group; checkboxes permit independent selections. <select> contains <option>; textarea accepts multiple lines.

For a text input, size indicates approximate visible width, whereas maxlength limits entered text length.

Neither means a server security bound. type="password" masks display but does not encrypt. submit submits; button has no submission action by itself. type=email/required give browser validation feedback; arbitrary clients can bypass it.

No GET/POST method universally means “parse one element at a time.”

Web Programming module

WHATWG HTML input standard

WEB10 | Frames, media and graphics

Legacy frameset/frame divides a window into separate documents; frameset replaced body in old frameset documents.

Named targets controlled which frame changed; independent scrolling could leave other frames stationary.

These elements are obsolete in modern HTML.

An iframe embeds a document within an ordinary page and needs appropriate permissions/sandboxing for untrusted content.

Modern audio/video elements embed supported media, often with controls and fallback content; embed/object can also embed supported resources.

Several tags can embed supported video resources; distinguish their semantics and browser support.

Old bgsound or Flash examples do not imply current browser support.

Canvas provides script-driven bitmap drawing via a context; SVG describes scalable shapes in the DOM.

For a simple canvas task, obtain getContext("2d"), then use fillRect(x,y,w,h); line/path methods require beginPath/moveTo/lineTo/stroke.

Preserve a text alternative where graphics convey information.

Web Programming module

WEB11 | CSS attachment and selector meaning

An external sheet uses <link rel="stylesheet" href="styles.css"> in head; an internal sheet uses style; an inline declaration uses an element's style attribute.

External CSS promotes reuse, consistent design and caching; it does not guarantee that every page becomes smaller or accessible.

A rule is selector { property: value; }.

Comments use /* ... */.

Type p selects paragraph elements, .notice selects a class, #logo an ID, h1,h2 groups selectors, nav ul selects descendants and nav>ul direct children.

Attribute selectors match properties such as [type="email"], pseudo-classes states such as :hover/:focus, and pseudo-elements parts such as ::before.

A selector chooses where declarations apply; it is not itself the property that sets a color.

Web Programming module

WEB12 | Resolve cascade, specificity and inheritance

First compare applicable declarations in their cascade origin, importance and layer context; then compare specificity and order within the relevant context.

In an ordinary same-origin/same-layer author example, an ID outweighs class/attribute/pseudo-class selectors, which outweigh type/pseudo-element selectors.

Compare counts lexicographically, not as an unlimited base-ten total.

Equally specific applicable rules use later source order.

An important declaration outranks a competing normal declaration in that context.

For <p id="intro" class="note">, #intro{color:blue} beats .note{color:red}.

A competing p{color:green!important} beats those normal declarations.

Inherited color/fonts apply when the element lacks a stronger specified value; margin/padding/borders do not normally inherit. :not() contributes the specificity of its argument rather than universally zero.

Web Programming module

Mozilla MDN specificity

WEB13 | Style properties and box calculations

color sets text color; background-color sets background; font-size size, font-family face, font-style italic, font-weight weight, text-align inline-content alignment, and text-decoration effects such as underline.

Margin lies outside the border; padding lies inside; border surrounds content/padding.

With content-box width200px, horizontal padding10px and borders2px, border-box width is224px, excluding margins.

With border-box width200px, content width is176px.

Shorthand values follow top/right/bottom/left: one applies everywhere, two vertical/horizontal, three top/horizontal/bottom, four each side.

Adjacent vertical block margins can collapse rather than add; padding does not collapse.

Non-replaced inline elements do not obey width/height like blocks.

A 1rem length uses root font size; font-size in em uses parent font size, whereas many other em lengths use the element's computed font size.

In historical markup, the body text attribute controlled default text color, while bgcolor controlled background.

Current CSS uses color and background-color instead.

Web Programming module

WEB14 | Layout, alignment and responsive design

Normal flow lays out blocks/inlines.

Flexbox arranges direct children on a main axis; justify-content follows that axis, align-items the cross axis.

With row direction in ordinary horizontal writing, cross-axis alignment is vertical.

Grid organizes rows and columns with explicit tracks, such as grid-template-columns:1fr 2fr. vertical-align applies to inline/table-cell contexts and does not center every block. align-content distributes tracks/lines when there is applicable extra space, not arbitrary single-child centering.

Relative positioning offsets a box while retaining its flow space; absolute removes it and uses its containing block, often established by a positioned ancestor; fixed commonly uses the viewport.

Responsive design combines flexible layout, images, viewport metadata and media queries, for example @media(max-width:600px){.cards{grid-template-columns:1fr}}.

A min-width768px query targets wider viewports, not mobile alone.

Browser compatibility requires feature testing and fallbacks; responsive size handling is a different issue.

Web Programming module

WEB15 | JavaScript syntax, inclusion and declarations

Use <script src="app.js"></script> for an external .js file; script content has no HTML closing-tag syntax inside its JavaScript source.

A deferred script runs after parsing and before DOMContentLoaded.

JavaScript originated at Netscape; it is distinct from Java and supports sophisticated applications through prototype-based objects and first-class functions.

Identifiers are case-sensitive and may start with letters, underscore or $, not digits or spaces. var has function/global scope; let/const are block-scoped. const prevents rebinding but not mutation of an object's contents. // comments end at line end; /* */ span a block.

Semicolons delimit statements, although automatic insertion exists.

Whitespace usually separates tokens but newlines can affect parsing, especially after return. "use strict" is a directive string, not a reserved statement keyword.

Parse syntax before computing outputs.

Web Programming module

WEB16 | Types, conversions and missing values

JavaScript primitives are undefined, null, Boolean, Number, BigInt, String and Symbol; objects include arrays, functions and dates.

Float is not a separate Number primitive type.

Arrays are objects, even though informal texts call them an array data type. typeof new Date() is "object"; typeof null is historically "object".

An unassigned declared variable or missing ordinary object property evaluates to undefined; directly reading an undeclared variable usually throws ReferenceError.

Strings use either quote style, with appropriate escaping. + can concatenate when a string is involved; arithmetic / converts ordinary numeric strings. "abc"/"def" produces NaN; nonzero Number divided by zero produces signed Infinity, while 0/0 produces NaN. parseInt("12.8",10) reads12, stopping at the dot; this is parsing, not rounding.

Global isNaN converts first; Number.isNaN does not.

Web Programming module

Mozilla MDN JavaScript data structures

WEB17 | Trace assignments, comparisons and postfix updates

Evaluate operators in their specified order and write down both value and type.

JavaScript has ==/!= with coercion and ===/!== without coercion; strict operators do not replace or remove the loose ones.

A condition ? x : y chooses one branch.

var a="20"; var b=a=30; assigns Number 30 to both, so a+b is 60, not 50. "1 plus 1 is "+1+1 concatenates left to right to "1 plus 1 is 11".

Postfix a++ yields the old value and then increments a. If a=10, a==a++ compares10 with10, then the body sees11.

With a=1, printing a-- then a prints10.

These JS evaluation rules are not portable to C/C++ expressions with unsequenced modifications.

Web Programming module

Mozilla MDN equality comparisons

Mozilla MDN increment operator

WEB18 | Math, string and array API microtraces

Math.abs(-9)=9, Math.floor(5.9)=5, and Math.sqrt(36)=6.

A JS string's substring(start,end) includes start and excludes end: "exitexam".substring(2,6) is "itex".

Indexes start at zero.

Array push/pop add/remove at the end; unshift/shift act at the beginning. slice returns a selected shallow copy; splice(start,deleteCount,...items) mutates by removal/insertion. reverse reverses in place. reduce combines sequentially from left to right; reduceRight combines sequentially from right to left with an accumulator, not simultaneous parallel execution.

For [1,2,3].reduceRight((a,v)=>a-v,0), accumulator values are -3,-5,-6.

An absent initial value uses the last present item first; an empty array without one throws.

Method spelling and case matter.

Web Programming module

Mozilla MDN reduceRight

Mozilla MDN Math

Mozilla MDN substring

WEB19 | Control flow, functions and errors

for(initialization;test;update) initializes once, checks before each iteration, then updates after each body. while checks before its body; do...while executes at least once. break exits and continue advances to the next iteration.

A loop from n=1 while n<7 prints a message only at n=5 when testing n%5==0.

function f(x){return x*2;} has a parameter and returns a value; functions can be passed/stored like other values.

Arrow syntax is compact but differs in this/constructor behavior.

Runtime argument limits mean "no limit whatsoever" is too absolute. try/catch handles thrown errors during execution; throw signals one; finally runs cleanup.

Parse-time syntax errors and logical errors are different. bool ans=true is not JavaScript declaration syntax; use let ans=true.

A literal type of expression is not the typeof operator.

Web Programming module

WEB20 | DOM selection, creation and safe updates

The DOM is a hierarchy of nodes representing the document. document.getElementById("x") returns the matching element or null; querySelector(".item") returns the first match, and querySelectorAll returns a static NodeList.

Other plural lookup methods can return live collections.

Guard missing elements and run code after the relevant markup exists.

To add a div: const el=document.createElement("div"); el.textContent="Hello"; document.body.appendChild(el);.

Creation alone does not insert it. createTextNode makes text; appendChild attaches an existing node. innerHTML parses/replaces descendants, so += reparses/replaces the prior descendant content too and can destroy listeners/state.

Use textContent for untrusted plain text. element.style.backgroundColor sets a style property; classList.add changes a class. getElementByld with a lowercase l replacing I is a different, nonexistent method.

Web Programming module

WEB21 | Events and validated submission

An event handler runs when an event occurs: click, input, keydown, mouseover, submit and load are examples. addEventListener("click",handler) registers a function without calling it immediately.

Multiple listeners may coexist; once removes one after use, capture selects an earlier propagation phase and passive promises no default cancellation.

Event bubbling is distinct from the browser's default action.

For submission, form.addEventListener("submit",e=>{if(!valid()){e.preventDefault();}}) stops the default only on failure.

Returning false from an addEventListener callback does not cancel it; return false works in an appropriate inline/onsubmit property handler whose caller uses that return.

For cancelable keyboard defaults, call preventDefault rather than assuming a different key event itself disappears.

Browser checks improve feedback; repeat validation and authorization on the server.

Web Programming module

Mozilla MDN addEventListener

WEB22 | Window, dialogs and browser properties

In a normal browser page, window exposes the browsing context and document is its document property. alert displays information, confirm returns a Boolean, and prompt requests text or returns null on cancellation. console.log logs diagnostics. location.reload() reloads the current page. screen.availHeight, screen.availWidth and screen.colorDepth have exact camelCase spelling; ColorsDepth is not the property.

window.onload is a handler property for the load event, typically after dependent resources finish.

Global let/const are not window properties like classic top-level var bindings. document.write is a method/function property; writing after document load can replace the document. document.bgColor is a legacy property; Document.bgcolor has wrong object/case.

Use document.body.style.backgroundColor for current styling.

Browser workers support concurrent computation, but DOM access normally remains on the page's main thread.

Web Programming module

Mozilla MDN Screen

WEB23 | PHP syntax, names and types

PHP means PHP: Hypertext Preprocessor.

Use <?php ... ?> in a .php file; a closing tag can be omitted at file end.

It can emit ordinary text and HTML.

PHP8 no longer accepts historical <script language="PHP"> delimiters; short <? depends on configuration, so use the full form.

A stray >> is not a closing PHP tag.

Variables begin with $, then a letter/underscore and subsequent letters/digits/underscores; $AGE18 works but $18AGE does not.

Names are case-sensitive.

Dynamic typing permits values of different types without a preceding type declaration, though optional declarations can constrain them.

Scalars include int,float,string,bool; arrays/objects are structured values and null means no value. var_dump displays type and value.

Constants cannot normally be redefined during execution.

Web Programming module

WEB24 | PHP output, strings and comparisons

echo is a language construct with no return value and can emit comma-separated expressions. print takes one expression and returns1.

They are two basic output constructs, not the only output mechanisms; printf, print_r and others exist.

Neither automatically inserts spaces/newlines.

Two prints of5 emit55; "I"."am" emitsIam.

Dot concatenates; + is arithmetic.

Double-quoted strings interpolate variables and escapes such as \n; single quotes normally preserve dollar names literally.

To emit literal $lfc use echo '$lfc'; or escape the dollar in double quotes.

String offsets start at zero: "maroon"[2] is r when the surrounding PHP is valid. strlen counts bytes; strpos can return integer0 or false, requiring ===false for failure. strrev reverses bytes and str_replace replaces matching content.

PHP 8==8.0 is true, but 8===8.0 is false because types differ; echo false emits an empty string, echo true emits1.

PHP sends markup as response characters; a browser then interprets it. echo "A
B" emits a tag in raw output but normally displays two text lines.

The invalid </br> markup depends on HTML error recovery, so specify raw versus rendered output.

Web Programming module

PHP manual: strings

PHP manual: echo

WEB25 | PHP scope, parameters and persistent locals

A variable outside a function has global scope.

A function’s locals and by-value parameters are separate bindings; use global $x or $GLOBALS["x"] to refer to a global explicitly.

Superglobals are available across scopes.

A static local retains its value across calls during the execution lifecycle; static storage is not simply another visibility scope.

An uninitialized static local begins as null, so echo emits no characters.

If global $i=10 and sample($i) increments/prints its by-value parameter, the caller’s $i stays10.

Printing before, during and after gives10,11,10 when commas are written by the source.

A global statement outside the function does not turn its value parameter into a reference.

Common “three scopes” teaching counts local, global and static, but state the taxonomy.

A static counter initialized0 then incremented on each call emits1 then2.

Web Programming module

PHP manual: variable scope

WEB26 | PHP arrays, loops and callable behavior

PHP arrays map integer/string keys to values.

Numeric arrays start at index0 unless keys are supplied: [20,12,23,19,18] has19 at index3.

Associative syntax uses a contiguous => operator: ["top"=>"black"] stores black under key"top".

Splitting it as = > is a parse error, not valid array syntax.

foreach($items as $key=>$value) iterates entries; for and while use tests/updates much like the corresponding JS forms. $sub-=$sub is $sub=$sub-$sub, not decrement.

For $sum=0; foreach([2,4] as $v){$sum+=$v;} the result is6.

A function can receive arguments, return a value, or produce output; classes/objects/variables also support server processing, so no single one of these mechanisms alone defines “dynamic PHP.” Use braces to make nesting explicit.

Web Programming module

WEB27 | Request data, server validation and injection

$_GET contains query parameters; $_POST contains supported form POST data; $_FILES upload metadata; $_SERVER request/environment details; $_COOKIE cookie values; $_SESSION session data after startup. $GLOBALS is the actual global-symbol array; $_GLOBAL is not the built-in name. $_REQUEST may combine GET/POST/cookie data according to configuration; it is not a database interface and has ambiguous precedence, so prefer the intended source. isset tests existence and non-null, not truth or valid syntax.

Read input, check required fields/type/length/range, authenticate and authorize, then process.

Bind SQL parameters to keep values separate from commands.

Encode output for HTML/attribute/JS context; textContent handles plain browser text.

A security test can detect SQL injection or XSS but does not itself fix either.

Client validation, encryption and a database connection do not replace these controls.

Web Programming module

WEB28 | Cookies and browser storage

Cookies are browser-held name/value data automatically attached to matching requests subject to domain, path, expiry, Secure and SameSite rules.

HttpOnly restricts script access; Secure requires secure transport.

They are not attached to every site indiscriminately.

PHP setcookie sends a response header, so set it before output unless buffering correctly defers output; $_COOKIE reflects received cookies.

localStorage persists origin-specific string values across sessions until removed/cleared; sessionStorage is scoped to a page session/tab.

Neither is automatically transmitted like matching cookies, but JavaScript can send stored data explicitly.

IndexedDB supplies structured local storage.

Thus all four are client storage techniques, and localStorage can participate in token-based session designs, with script-access risks.

Storage size limits vary.

Being logged in after reopening may use a persistent session/remember token or browser restoration; storage alone does not prove server acceptance.

Web Programming module

Mozilla MDN localStorage

WEB29 | Server sessions and the authorization trust path

session_start() starts/resumes PHP session handling and makes $_SESSION available.

Normally the browser holds an opaque session identifier while data resides in server-controlled storage.

A counter initializes missing $_SESSION["views"] to1 and increments on later requests. unset removes one entry; destroying stored session data is distinct from clearing the browser cookie/current variables, so logout must handle the full lifecycle.

A client cookie logged_in=1 is attacker-controlled.

If the server trusts it alone, authorization can be bypassed.

If the server checks a valid session ID and its server-side user/role on every protected action, editing that unrelated flag does not grant rights.

HTTPS does not make client-generated claims trustworthy.

Determine which server-side check is performed before judging whether a session-handling design is secure or vulnerable.

Protect session IDs, regenerate after login, expire appropriately and avoid putting sensitive tokens in URLs.

Web Programming module

WEB30 | Uploads, files and storage safety

Uploads ordinarily use method="post", enctype="multipart/form-data" and a named file input. accept hints to the browser which file types to offer; it does not enforce the server's policy.

Validate server-side size/content, inspect upload errors, use the temporary upload path, choose a safe generated destination and use move_uploaded_file.

Avoid executable storage and trusting a client filename/extension.

file_exists tests files or directories; is_file narrows to regular files. fopen returns a stream or false: r reads an existing file, r+ reads/writes without truncation, w truncates/creates for writing, w+ adds reading, a appends/creates, and x exclusively creates. fclose releases it. readfile emits content and returns a byte count, so echo readfile(...) also prints that count.

Check failures before reading/writing and constrain paths to permitted locations.

Web Programming module

WHATWG HTML input standard

WEB31 | Database connections, TLS and actual APIs

MySQL is a relational DBMS; a database persists structured application data and supports concurrent clients.

PHP8 uses MySQLi or PDO; the old mysql_* extension was removed, so migrate older snippets to a supported API. mysqli_connect(host,user,password,database,port,socket) opens a connection; selecting a database can be deferred, and defaults make several arguments technically optional.

Do not reduce the full signature to exactly one optional field.

Configure MySQLi TLS options through mysqli_init(), mysqli_ssl_set(...) and mysqli_real_connect(...) with the required verification configuration.

mysqli_ssl_connect(), pdo_connect_ssl(), mysql_ssl_connect() and secure_connect() are not substitutes for the MySQLi TLS configuration sequence above.

PDO uses an appropriate driver DSN/options instead.

MySQLi procedural functions, object methods and PDO methods have distinct signatures; affected_rows() alone is not a generic PHP function.

Web Programming module

PHP manual: mysqli::ssl_set

PHP manual: affected_rows

WEB32 | SQL operations, result rows and safe rendering

SELECT retrieves rows; INSERT adds; UPDATE ... SET changes values; DELETE removes rows.

WHERE limits affected/matching records; omitting it from DELETE ordinarily targets every row, subject to constraints, permissions and errors.

CREATE DATABASE/TABLE defines storage; a primary key identifies rows.

AUTO_INCREMENT generates numeric values but gaps and explicit values are possible; do not promise gapless consecutive business numbering.

mysqli_query returns a result object for successful result-producing queries, true for other successful statements, or failure/exception depending on settings.

Fetch associative rows with mysqli_fetch_assoc, positional rows with mysqli_fetch_row, or objects with mysqli_fetch_object. mysqli_num_rows counts a result's rows; mysqli_affected_rows($conn) counts changed/inserted/deleted rows under its options; mysqli_insert_id($conn) reports a generated AUTO_INCREMENT value from the connection.

Escape rendered cell text; use prepared statements for submitted values; mysqli_close closes the connection.

Web Programming module

PHP manual: affected_rows

WEB33 | PHP includes, objects and dates

include evaluates another file in the current scope and warns on failure; require fails critically.

In PHP8 a failed require raises Error, which can be caught; uncaught it stops normal execution.

The source's E_COMPILE_ERROR description belongs to older PHP. include_once/require_once avoid repeated inclusion.

Do not include a user-selected arbitrary path.

A class defines properties/methods; new creates an object, $obj->field accesses a property, and $obj->method() calls a method. public is externally accessible, private is class-restricted, and a constructor initializes an instance.

Use $this inside instance methods.

DateTime/DateTimeZone represent time and zone; diff returns a DateInterval and DatePeriod iterates intervals.

Time calculations require explicit zones and calendar assumptions, not an assumed server-local timezone.

Web Programming module

PHP manual: require

WEB34 | AJAX and response handling

AJAX means Asynchronous JavaScript and XML, but responses can be JSON, text, HTML or XML.

It is a technique, not a language, standalone framework or mandatory XML protocol.

Configure an XHR with open("GET",url,true), register response/error handlers and call send(). onreadystatechange may run several times; readyState4 means completion, after which status and data must be checked.

A callback is a function invoked later by the runtime/API.

The browser invokes its response handler when the response is available; the remote server does not directly execute browser code.

With fetch, await fetch(url) obtains a response, then check response.ok and parse await response.json().

Handle network/parse errors.

Starting an asynchronous operation does not make its result synchronously available to the next statement.

Avoid overwriting a newer search result with an older response that arrives late.

Web Programming module

WEB35 | Polling, WebSockets and data formats

Short polling repeatedly requests updates.

Long polling holds one client request until an event or timeout, returns a response, then the client reconnects; it is not continuous unsolicited streaming on that same finished response.

WebSocket establishes a persistent bidirectional message channel after its handshake.

Either HTTP-based AJAX or WebSockets can support nonblocking user interaction and carry JSON, so generic questions may be nonexclusive.

JSON represents objects, arrays, strings, numbers, Booleans and null; keys/strings use double quotes, and functions/undefined are not JSON values.

XML is another valid interchange format.

A REST API can return either.

Select transport by latency, direction, connection cost and recovery requirements, not merely the occurrence of “asynchronous.” Validate incoming messages and enforce authorization per operation.

Web Programming module

WEB36 | Node server, event loop and scaling

A minimal CommonJS Node server can use const http=require("node:http"); http.createServer((req,res)=>{res.writeHead(200,{"Content-Type":"text/plain"});res.end("Hello");}).listen(8000);.

The request listener runs for incoming requests; end completes the response.

Define every variable used in startup logging, including hostname, before using it.

Node schedules JavaScript callbacks through an event loop and uses OS/worker facilities for asynchronous work.

Concurrency means overlapping progress; parallelism means execution at the same instant.

One event-loop thread can handle many waiting requests, but long CPU loops or synchronous I/O block its callbacks.

Workers/processes and a load balancer distribute work across server instances; they do not automatically coordinate shared data or session state.

Multiple clients, or one client with overlapping requests, can both create concurrency.

Web Programming module

Node.js: do not block the event loop

WEB37 | Races, locks and transaction invariants

Suppose two requests read a stock count1, both decide it is available and both write0: two sales consumed one item.

Correctness depends on interleaving, even if each callback has only a short synchronous body.

A mutex protects one critical section; semaphores can coordinate bounded access; spinlocks busy-wait and are unsuitable for blocking a browser/event-loop thread.

Several listed mechanisms can synchronize, so mutex is not the only possible answer.

In a relational database use an atomic conditional UPDATE stock SET qty=qty-1 WHERE id=?

AND qty>0, check affected rows, and combine related writes in an appropriate transaction.

A row lock or serializable transaction can protect multi-step logic; handle rollback/retry on conflicts.

A process-local lock does not cover all load-balanced servers.

Idempotency tokens with unique storage constraints prevent a retried payment from becoming a duplicate transaction.

PostgreSQL transaction isolation

WEB38 | Frameworks and version-aware setup

A framework supplies conventions and reusable structure; a library supplies callable capabilities. jQuery simplifies DOM operations; React builds component interfaces; Angular is a component platform, distinct from older AngularJS; Vue also builds interactive interfaces.

Each can contribute to a responsive interface, so those generic descriptions do not identify one unique framework.

React props pass inputs; state records changing component data; JSX needs transformation rather than being ordinary browser JavaScript syntax.

Bootstrap supplies layout/components.

The jQuery→Popper→Bootstrap dependency order is version4-era guidance; Bootstrap5 drops required jQuery.

Angular CLI commands include ng new, ng serve, ng build, ng generate and ng test, with e2e support depending on configuration. Use ng serve to start the development server.

Create React App is deprecated; recognize it as historical tooling rather than a universal current recommendation.

Web Programming module

Bootstrap5 migration

React installation

React: passing props

WEB39 | Manage content, performance and revisions

A CMS gives editors content creation, publishing, roles and media management without editing every source file.

A database stores persistent facts; it is not itself the whole server application.

A VCS records source history, branches and merges to support review/collaboration; deployment publishes a chosen build and is a separate operation.

SEO improves discoverability through meaningful content, headings, metadata, accessible links and crawlable structure; keyword metadata alone does not guarantee ranking.

Caching reuses a representation/result and requires validity/invalidation rules.

Server caching can hold rendered pages or expensive query results; never share private user content under an unsafe common key.

Minification removes unnecessary source characters; bundling combines files; compression changes transfer encoding; preloading fetches anticipated resources early.

Measure performance because excess preloading/bundling may compete with critical work.

Responsive design and cross-browser compatibility need separate testing.

Web Programming module

WordPress features

Google Search Central SEO Starter Guide

Pro Git: About Version Control

WEB40 | Assemble and evaluate an end-to-end feature

For a course-search page, write semantic headings/navigation and a labeled search form.

Use a responsive grid with a single-column fallback, then register a submit handler after parsing.

Prevent default only when implementing asynchronous submission; encode search values in the query and send GET /courses?q=.... Show loading and useful error feedback without inserting untrusted HTML.

The server validates query length/type, authenticates where required, runs a parameterized SELECT with a bounded result size, and returns JSON with an appropriate status/content type.

The client checks status, parses data and builds nodes with textContent.

For an enrollment action use POST with CSRF controls, server authorization, capacity checks and a transaction; show success only after commit.

Evaluate empty/bad input, access denied, network failure, delayed responses, repeated clicks, simultaneous enrollments, narrow screens and keyboard use.

This integrates all four blueprint outcomes beyond tag recall.

Web Programming module

WEB41 | Validate syntax, context and output

A trace requires syntactically valid source, known initial values, execution context and output representation. x=4+"4"5 is a syntax error because an operator is missing; bool ans=true and type of new Date() are syntax errors too.

PHP with >> instead of ?> may fail parsing; analyze output only after checking the closing syntax.

The PHP array operator is =>; splitting it as = > is invalid syntax.

Distinguish response text from browser rendering: PHP may emit markup that the browser interprets, and visual line breaks need not appear in a plain-text choice.

For the assignment example above, the correct output is 60 even if it is absent from the choices.

Multiple storage techniques, interface libraries, script-capable languages and cryptographic channels often satisfy broad descriptions.

State the missing distinction before selecting an answer when the question admits several interpretations.

With x=4+"4";, JavaScript produces 44; x=4+"4"5 is invalid because an operator is missing.

The PHP string-offset example closes with ?> and prints r; using >> instead introduces a syntax error.

Check the complete code and closing delimiters before tracing output.

Apply each output prediction to the exact program being analyzed.

Web Programming module

WEB-S01 | Java and JavaScript object-oriented models

Java organizes object-oriented programs using classes, objects, inheritance and interfaces. A class describes the state and behavior of its instances; inheritance relates a subclass to a superclass, and an interface specifies operations a type agrees to provide.

Oracle Java Tutorials: Object-Oriented Programming Concepts

WEB-S02 | Form values and result destinations

For a text input, the value attribute supplies its initial text value; it does not specify the control's width. The type attribute selects the kind of input control.

A form's target identifies the window or frame in which the submitted result is displayed. The action identifies the processing URL, so changing the destination context is different from changing the handler.

In an input element, type="text" creates text entry, type="checkbox" an independently selectable checkbox, type="radio" a radio choice, and type="submit" a submit button. A hidden input carries a value without displaying an editable text field; it is not a password-masking control.

HTML Standard: The input element

Web Programming module · page 21

WEB-S03 | Message display versus response collection

alert(message), confirm(message) and prompt(message) all present a supplied message. Alert is a notification; confirm additionally asks for an OK/Cancel decision, while prompt additionally offers a text field. A general message-display description therefore differs from an information-only notification.

Web Programming module · page 46

Web Programming module · page 47

WEB-S04 | Database API namespaces and affected rows

PDOStatement::rowCount() is a method on a PDO statement object that reports affected rows for its DELETE, INSERT or UPDATE statement. It is not a procedural MySQLi function, and its behavior for SELECT result counts is driver-dependent rather than a portable row-count guarantee.

With an initialized PDO connection, $pdo->query($sql) executes an SQL statement without placeholders and returns a PDOStatement for processing its results. The connection's configured driver determines the database; the method name alone does not mean MySQL.

MySQL is an open-source SQL database management system. Its relational data-management role is distinct from the PHP API used to connect to it.

The procedural MySQLi query form is mysqli_query($conn, $sql), using a MySQLi connection and the correctly spelled function name. The pg_query function belongs to PHP's PostgreSQL extension; it is not a MySQL connection API.

PHP Manual: PDOStatement::rowCount

PHP Manual: PDO::query

Web Programming module · page 78

PHP Manual · mysqli_query

PHP Manual · pg_query

WEB-S05 | JavaScript execution, statement forms and error categories

JavaScript implementations may interpret or compile source; interpretation and compilation are execution strategies rather than competing language identities.

An if statement conditionally executes code. The legacy with statement temporarily extends name lookup and is forbidden in strict mode. A debugger statement may invoke an active debugging facility. These are keyword-introduced statement forms, unlike the quoted use-strict directive.

A syntax error prevents source from being parsed successfully. A runtime error arises while executing a program. A logical error is a defect in the algorithm or expression that can produce an incorrect result without throwing an exception.

Web Programming module · page 44

ECMAScript: Statements and declarations

Mozilla MDN: Troubleshooting JavaScript

WEB-S06 | PHP statement boundaries and string escapes

A complete ordinary PHP assignment statement ends with a semicolon. A string's closing delimiter must match its opening delimiter; a single quote can appear inside a double-quoted string without ending it.

In double-quoted PHP strings, backslash-n represents a line feed and backslash-r represents a carriage return. The written sequences are \n and \r; forward-slash sequences such as /n are ordinary text. A backslash before a dollar sign preserves that dollar sign rather than starting variable interpolation.

The basic print and echo constructs can be written with or without parentheses when their syntax permits it. Parentheses do not add spaces or change the value of the emitted argument.

PHP Manual: Strings

Web Programming module · page 62

Web Programming module · page 63

WEB-S07 | Default text rendering and legacy frame fallbacks

In the browser's default presentation, an i element typically displays its text in italics. This is a rendering convention: its text semantics are distinct from the stress emphasis represented by em, and CSS may override appearance.

A legacy frameset document could include noframes content for user agents that did not support frames or were configured not to display them. Frame support was therefore not universal even in that historical model.

Web Programming module · page 15

Web Programming module · page 20

WEB-S08 | Numeric functions and NaN checks

For finite numeric arguments, Math.abs(x) gives the nonnegative magnitude, Math.floor(x) gives the greatest integer no larger than x, and Math.sqrt(x) gives the nonnegative square root for nonnegative x. These operations differ from ordinary nearest-integer rounding; floor moves toward negative infinity.

Global isNaN(value) tests whether conversion to Number produces NaN. Number.isNaN(value) tests specifically for the Number value NaN without coercion. NaN is itself a special numeric value, not a callable checking function.

ECMAScript: Math object functions

MDN · isNaN

WEB-S09 | Destroying server-side session data

session_destroy() removes the data associated with the current PHP session. It does not itself unset every current script variable or delete the browser's session cookie; destroying stored data, clearing $_SESSION and expiring the identifier cookie are separate parts of a complete logout.

Web Programming module · page 77

PHP Manual: session_destroy

WEB-S10 | HTTP response status classes

The first digit of an HTTP status code identifies its class: 2xx means successful handling, 3xx means redirection or further action, 4xx indicates a client-error condition, and 5xx indicates a server-error condition. Classifying a code is different from identifying the precise condition represented by its last two digits.

RFC 9110: Status codes

WEB-S11 | Authentication versus authorization

Authentication verifies a claimed identity. Authorization checks whether a particular entity is permitted to perform a requested action. Establishing identity does not grant every permission, and neither check alone repairs unsafe SQL construction or unsafe output encoding.

OWASP Authentication Cheat Sheet

OWASP Authorization Cheat Sheet

WEB-S12 | Metadata purposes

The keywords metadata name supplies terms relevant to the page, while description supplies a page summary. Robots metadata supplies crawler indexing or search-presentation instructions. These roles differ from mobile viewport control.

HTML Standard: Standard metadata names

Google Search Central: Robots meta tag

WEB-S13 | Remainder and logical conjunction

For nonnegative integers n and positive d, n % d gives the remainder after division; zero tests divisibility. With Boolean operands, && is true only when both operands are true; it is a logical operator, not a comment delimiter.

Mozilla MDN: Remainder operator

Mozilla MDN: Logical AND

WEB-S14 | Application protocols and underlying communication

HTTP defines web application requests, response status codes and representations. An HTTP exchange can use underlying TCP connections and IP addresses; naming that network stack does not identify the application-level method or status semantics.

FTP is a file-transfer application protocol. SMTP transports electronic mail. These service roles differ from the HTTP resource request/response semantics used by a web browser. HTTPS adds TLS protection to HTTP rather than replacing its resource semantics.

RFC 9110 · HTTP Semantics, §§3 and 4

RFC 959 · File Transfer Protocol

RFC 5321 · Simple Mail Transfer Protocol