

FINAL TERM PREPARATION MCQS

CHAPTER WISE MCQS (22-45)

(VU UPDATES TECH)

☐ **Week 08 – Touch Interaction & Touch UI**

(15 MCQs)

1. Touch interaction is mainly designed for:
 - A) Keyboard devices
 - B) Mouse devices
 - C) **Touchscreen devices**
 - D) Voice devices
2. Touch UI is best suited for:
 - A) Desktop PCs
 - B) Servers
 - C) **Mobile devices**
 - D) Mainframes
3. A touch event is generated when:
 - A) Mouse moves
 - B) Key is pressed
 - C) **User touches screen**
 - D) Screen refreshes
4. Touch UI should be:
 - A) Complex
 - B) **Simple**
 - C) Text-heavy
 - D) Command-based
5. Touch targets should be:
 - A) Very small
 - B) **Large enough to tap easily**
 - C) Hidden
 - D) Overlapping
6. “Don’t do anything” rule means:
 - A) Ignore user

- B) **Avoid accidental actions**
 - C) Block UI
 - D) Disable input
7. Click delay exists because of:
- A) Hardware issue
 - B) **Double-tap detection**
 - C) Network delay
 - D) Browser bug
8. Touch UI should avoid:
- A) Buttons
 - B) Icons
 - C) **Hover effects**
 - D) Colors
9. Touch gestures include:
- A) Click
 - B) Hover
 - C) **Tap, swipe, pinch**
 - D) Scroll bar
10. Supporting touch means:
- A) Mouse only
 - B) Keyboard only
 - C) **Touch-friendly UI**
 - D) Voice control
11. Touch interfaces reduce need of:
- A) Screen
 - B) **Keyboard & mouse**
 - C) Internet
 - D) Battery
12. Touch interaction improves:
- A) Code size
 - B) **User experience**
 - C) Memory usage
 - D) Compilation
13. Multi-touch allows:
- A) One finger only
 - B) **Multiple fingers input**
 - C) Mouse clicks
 - D) Keyboard shortcuts
14. Touch UI is optimized for:
- A) Precision
 - B) **Ease of use**
 - C) Complexity
 - D) Accuracy only
15. Touch UI follows:
- A) Hardware rules
 - B) Network rules

- C) **Human interaction rules**
 - D) OS rules
-

☐ **Week 09 – Mouse Emulation & Touch Events**

(15 MCQs)

1. Mouse emulation means:
 - A) Replacing mouse
 - B) **Simulating mouse using touch**
 - C) Removing mouse
 - D) Enhancing mouse
2. Touch can emulate:
 - A) Keyboard
 - B) Monitor
 - C) **Mouse events**
 - D) CPU
3. Synthesizer application is used to:
 - A) Store data
 - B) **Generate sound**
 - C) Compile code
 - D) Parse XML
4. Touchstart event occurs when:
 - A) Finger leaves screen
 - B) **Finger touches screen**
 - C) Screen scrolls
 - D) Page loads
5. Touchend event occurs when:
 - A) Touch starts
 - B) **Touch is released**
 - C) Touch moves
 - D) Screen refreshes
6. Touchmove event fires when:
 - A) Tap occurs
 - B) **Finger moves on screen**
 - C) Page loads
 - D) Click happens
7. Mouse emulation helps:
 - A) Keyboard users
 - B) **Legacy applications**

- C) Servers
 - D) Databases
 - 8. Touch events are part of:
 - A) CSS
 - B) **JavaScript**
 - C) HTML only
 - D) PHP
 - 9. Touch event object contains:
 - A) Only text
 - B) **Touch coordinates**
 - C) Images
 - D) Styles
 - 10. Touch APIs are mainly used in:
 - A) Desktop apps
 - B) **Mobile web apps**
 - C) Servers
 - D) Databases
 - 11. Emulation ensures:
 - A) Faster internet
 - B) **Backward compatibility**
 - C) Better graphics
 - D) Security
 - 12. Mouse emulation is needed when:
 - A) Touch only UI exists
 - B) **Old apps depend on mouse**
 - C) Keyboard used
 - D) Voice input used
 - 13. Touch events improve:
 - A) Coding speed
 - B) **Interactivity**
 - C) File size
 - D) Memory
 - 14. Touch-based synthesizer uses:
 - A) Keyboard input
 - B) Mouse clicks
 - C) **Touch gestures**
 - D) Voice
 - 15. Touch events are handled using:
 - A) HTML tags
 - B) CSS
 - C) **Event listeners**
 - D) Database
-

□ Week 10 – Pointer Events & Polyfill

(15 MCQs)

1. Pointer events unify:
 - A) Keyboard & mouse
 - B) **Mouse, touch & pen**
 - C) UI & UX
 - D) HTML & CSS
2. Pointer event object represents:
 - A) Keyboard input
 - B) **Pointing device**
 - C) Network request
 - D) Screen size
3. Pointer events simplify:
 - A) UI styling
 - B) **Input handling**
 - C) Database queries
 - D) Networking
4. Polyfill is used to:
 - A) Remove bugs
 - B) **Provide missing features**
 - C) Speed up code
 - D) Compress files
5. Polyfill library helps:
 - A) New browsers only
 - B) **Older browsers**
 - C) Servers
 - D) Databases
6. Pointerdown occurs when:
 - A) Pointer released
 - B) **Pointer pressed**
 - C) Pointer moves
 - D) Page loads
7. Pointerup occurs when:
 - A) Pointer pressed
 - B) **Pointer released**
 - C) Pointer moves
 - D) Scroll happens
8. Pointermove fires when:
 - A) Click happens
 - B) **Pointer moves**
 - C) Page loads
 - D) Zoom occurs

9. Pointer events reduce need of:
- A) Touch events
 - B) Mouse events
 - C) **Multiple input handlers**
 - D) HTML
10. PointerId identifies:
- A) Browser
 - B) **Input device**
 - C) OS
 - D) Page
11. Pointer events are part of:
- A) HTML5
 - B) CSS3
 - C) **W3C specification**
 - D) PHP
12. Polyfill ensures:
- A) Security
 - B) **Compatibility**
 - C) Performance
 - D) Accuracy
13. Pointer events support:
- A) Single input
 - B) **Multiple inputs**
 - C) Keyboard only
 - D) Mouse only
14. Pointer events are handled via:
- A) HTML attributes
 - B) **JavaScript**
 - C) CSS
 - D) XML
15. Pointer events are better because they are:
- A) Complex
 - B) **Unified**
 - C) Slower
 - D) Limited

☐ **Week 11 – Pointer Code Examples & Guidelines**

(15 MCQs)

1. Pointer guidelines focus on:
 - A) Code length
 - B) **User interaction**
 - C) Database design
 - D) Security
2. Pointer code should be:
 - A) Device specific
 - B) **Device independent**
 - C) OS specific
 - D) Hardware specific
3. Good pointer handling avoids:
 - A) Responsiveness
 - B) **Conflicts between inputs**
 - C) Performance
 - D) Compatibility
4. Pointer capture allows:
 - A) Losing pointer
 - B) **Tracking pointer outside element**
 - C) Ignoring input
 - D) Reset UI
5. Pointer guidelines recommend:
 - A) Hover dependency
 - B) **Avoid hover-only actions**
 - C) Small buttons
 - D) Hidden UI
6. Code examples show how to:
 - A) Design UI
 - B) **Handle pointer events**
 - C) Write SQL
 - D) Build servers
7. Pointer cancel occurs when:
 - A) Input completes
 - B) **System interrupts input**
 - C) User clicks
 - D) Page reloads
8. Pointer guidelines improve:
 - A) Compilation
 - B) **Accessibility**
 - C) Memory usage
 - D) Network speed
9. Pointer capture method is:
 - A) `releasePointerCapture`
 - B) **`setPointerCapture`**
 - C) `getPointer`
 - D) `lockPointer`

10. Pointer guidelines emphasize:
- A) Hardware usage
 - B) **Consistent behavior**
 - C) Complex UI
 - D) Graphics
11. Pointer examples use:
- A) HTML only
 - B) CSS only
 - C) **JavaScript**
 - D) PHP
12. Pointer events work well with:
- A) Touch only
 - B) Mouse only
 - C) **All pointing devices**
 - D) Keyboard only
13. Pointer guidelines help developers to:
- A) Write longer code
 - B) **Avoid bugs**
 - C) Increase size
 - D) Reduce UI
14. Pointer event handling is:
- A) Optional
 - B) **Important for modern apps**
 - C) Deprecated
 - D) Rare
15. Pointer guidelines focus on:
- A) Server-side logic
 - B) **Client-side interaction**
 - C) Databases
 - D) APIs

☐ Week 12 – Input Introduction & Keyboard Input

(15 MCQs)

1. Input element is used to:
- A) Display text
 - B) **Accept user data**
 - C) Style page
 - D) Link pages

2. Keyboard input allows:
 - A) Mouse interaction
 - B) Touch interaction
 - C) **Text entry**
 - D) Voice commands
3. Input types are defined using:
 - A) id attribute
 - B) name attribute
 - C) **type attribute**
 - D) class attribute
4. Keyboard input improves:
 - A) UI colors
 - B) **Accessibility**
 - C) Graphics
 - D) Speed
5. Text input is created using:
 - A) <text>
 - B) <input>
 - C) **<input type="text">**
 - D) <textbox>
6. Keyboard input is essential for:
 - A) Touch devices only
 - B) **Form filling**
 - C) Graphics
 - D) Video
7. Input focus means:
 - A) Cursor hidden
 - B) **Ready to receive input**
 - C) Input disabled
 - D) Page refresh
8. Keyboard navigation helps:
 - A) Gamers
 - B) Designers
 - C) **Disabled users**
 - D) Servers
9. Input fields should be:
 - A) Hidden
 - B) **Clearly visible**
 - C) Disabled
 - D) Overlapping
10. Keyboard events include:
 - A) onclick
 - B) **keydown**
 - C) onhover
 - D) onscroll

11. Keyboard input works without:

- A) Screen
- B) **Mouse**
- C) Browser
- D) OS

12. Input introduction focuses on:

- A) Styling
- B) **User interaction**
- C) Databases
- D) APIs

13. Keyboard input is part of:

- A) CSS
- B) **HTML & JavaScript**
- C) XML
- D) SQL

14. Input elements are mostly used in:

- A) Tables
- B) **Forms**
- C) Lists
- D) Images

15. Keyboard input supports:

- A) Only numbers
- B) **Various data types**
- C) Images
- D) Audio

☐ **Week 13 – Validation & Older Browsers**

(15 MCQs)

1. Validation ensures:

- A) Speed
- B) **Correct data input**
- C) Styling
- D) Storage

2. Regular expressions are used for:

- A) Styling
- B) **Pattern matching**
- C) Database access
- D) Security

3. HTML5 validation works on:

- A) Server side
- B) **Client side**

- C) Database
- D) OS
- 4. Input validation reduces:
 - A) Performance
 - B) **Errors**
 - C) UI
 - D) Accessibility
- 5. Older browsers may not support:
 - A) HTML
 - B) CSS
 - C) **HTML5 input types**
 - D) Images
- 6. Wrapping input in label improves:
 - A) Styling
 - B) **Accessibility**
 - C) Speed
 - D) Memory
- 7. TelecoIon URL is used for:
 - A) Email
 - B) **Phone calls**
 - C) Images
 - D) Downloads
- 8. Two important features of input are:
 - A) Size & color
 - B) **Validation & type**
 - C) Speed & memory
 - D) Border & margin
- 9. Regex is short for:
 - A) Regular example
 - B) **Regular expression**
 - C) Registered expression
 - D) Runtime expression
- 10. Validation can be:
 - A) Manual only
 - B) **Automatic**
 - C) Disabled always
 - D) Server only
- 11. Older browsers need:
 - A) Polyfill
 - B) **Fallback solutions**
 - C) Updates
 - D) Replacement
- 12. Input validation enhances:
 - A) UI design
 - B) **Data integrity**

- C) Speed
 - D) Graphics
 - 13. Label tag is associated using:
 - A) id
 - B) name
 - C) **for attribute**
 - D) type
 - 14. Client-side validation happens before:
 - A) Page load
 - B) **Form submission**
 - C) Server response
 - D) Storage
 - 15. Validation mainly improves:
 - A) Design
 - B) **User experience**
 - C) Compilation
 - D) Rendering
-

☐ **Week 14 – Sensors & Media Access**

(15 MCQs)

- 1. Mobile sensors detect:
 - A) Files
 - B) **Environmental data**
 - C) Network
 - D) Memory
- 2. Camera access allows:
 - A) Storage
 - B) **Capturing images/video**
 - C) Browsing
 - D) Coding
- 3. getUserMedia() is used for:
 - A) File upload
 - B) **Media capture**
 - C) Styling
 - D) Networking
- 4. Sensors include:
 - A) Keyboard
 - B) Mouse
 - C) **Accelerometer**
 - D) CPU

5. Capture extension is used to:
 - A) Delete media
 - B) **Specify capture source**
 - C) Encrypt media
 - D) Resize images
6. Camera access requires:
 - A) Auto permission
 - B) **User permission**
 - C) Server permission
 - D) OS permission only
7. Sensor data is mostly used in:
 - A) Desktop apps
 - B) **Mobile apps**
 - C) Servers
 - D) Databases
8. Drawback of capture extension is:
 - A) Speed
 - B) **Limited browser support**
 - C) Accuracy
 - D) Quality
9. Navigator object provides:
 - A) UI
 - B) **Device capabilities**
 - C) Styling
 - D) Database
10. getUserMedia() returns:
 - A) File
 - B) **Media stream**
 - C) Image
 - D) URL
11. Sensors enhance:
 - A) Coding
 - B) **Interactivity**
 - C) Compilation
 - D) Styling
12. Camera access is part of:
 - A) CSS
 - B) **HTML5 API**
 - C) XML
 - D) SQL
13. Sensor APIs are:
 - A) Unsafe
 - B) **Permission based**
 - C) Automatic
 - D) Deprecated

14. Media capture is mainly used for:

- A) Gaming
- B) **Video calls**
- C) Printing
- D) Storage

15. Sensors help create:

- A) Static apps
- B) **Context-aware apps**
- C) Simple pages
- D) Reports

☐ **Week 15 & 16 – Review (Final Focus)**

(15 MCQs)

1. Review focuses on:

- A) New topics
- B) **Revision of concepts**
- C) Assignments
- D) Labs

2. Final exam includes:

- A) Only coding
- B) Only theory
- C) **Mixed concepts**
- D) Projects

3. Touch, pointer & input relate to:

- A) Backend
- B) **User interaction**
- C) Database
- D) Networking

4. Pointer events combine:

- A) UI & UX
- B) **Multiple input types**
- C) CSS & HTML
- D) Server & client

5. Validation is important for:

- A) Speed
- B) **Correct data**
- C) Styling
- D) Images

6. Sensors are mainly used in:

- A) Servers
- B) **Mobile devices**

- C) Desktops
- D) Databases
- 7. getUserMedia() relates to:
 - A) Validation
 - B) **Camera & media**
 - C) Input text
 - D) Pointer
- 8. Accessibility improves for:
 - A) Developers
 - B) **All users**
 - C) Servers
 - D) Browsers
- 9. Review chapters help in:
 - A) Skipping syllabus
 - B) **Exam preparation**
 - C) Coding only
 - D) Assignments
- 10. HTML5 enhances:
 - A) Backend
 - B) **Frontend capabilities**
 - C) Database
 - D) OS
- 11. Pointer guidelines reduce:
 - A) Code
 - B) **User errors**
 - C) Memory
 - D) Speed
- 12. Input validation should be:
 - A) Optional
 - B) **Mandatory**
 - C) Rare
 - D) Avoided
- 13. Sensors require:
 - A) Automatic access
 - B) **User consent**
 - C) No permission
 - D) Server approval
- 14. Final exam is based on:
 - A) Assignments only
 - B) Quizzes only
 - C) **Lectures & handouts**
 - D) Projects only
- 15. CS420 focuses on:
 - A) Databases
 - B) Networking

- C) **User interaction & web technologies**
- D) AI

VU
Updates Tech