

UNIT — I HCI INTRODUCTION (08 Hours)

The Human: I/O channels — Memory — Reasoning and problem solving; The computer: Devices — Memory — processing and networks; Historical evolution of HCI; Interaction: Models — frameworks — Ergonomics — styles — elements — interactivity- Paradigms.

Textbook 1: Chapter 1 to 4

Definition:

HCI is about how humans interact with computers and how to design systems that are easy, efficient, and enjoyable to use.

“How many of you use apps like Instagram, Swiggy, or gaming apps?”

“Did you ever uninstall an app because it was confusing or difficult to use?”

“That’s exactly where *Human Computer Interface* comes in—it helps design systems that are not just functional but also user-friendly and enjoyable.”

Bad UI	Good UI
Form refreshes entire page after submission → user loses data if error occurs.	Real-time validation with inline error messages.
Date entry: Users must type in DD/MM/YYYY manually.	Date picker calendar for easy selection.
No default country code for phone numbers.	Auto-detects country code based on location.

1. Bad vs. Good UI → Restaurant Menu Analogy

- **Bad UI:** Imagine entering a restaurant where the menu is messy—items are unorganized, no prices are visible, and waiters are hard to find. You feel confused and frustrated.
- **Good UI:** In a well-designed restaurant, the menu is clear, items are categorized (Starters, Main Course, Desserts), prices are visible, and ordering is easy.
- **Lesson:** A confusing interface is like a confusing restaurant menu. A good interface gives clarity, structure, and comfort.

2. Navigation → Shopping Mall Analogy

- **Bad UI:** A shopping mall without maps or directions—customers wander without knowing where stores are.
- **Good UI:** A mall with clear maps, signboards, and “You are here” indicators helps customers reach their destination quickly.
- **Lesson:** Good navigation in UI is like signboards in malls—guides users efficiently.

3. Error Messages → Road Sign Analogy

- **Bad UI:** Driving on a road with no warning signs for sharp turns—high chance of accidents.
- **Good UI:** Clear signs before every turn help drivers slow down and stay safe.
- **Lesson:** Error messages and confirmations in UI are like road signs—they prevent mistakes.

4. Consistency → Car Controls Analogy

- **Bad UI:** In one car, the accelerator is on the left; in another, it's on the right. Confusing and dangerous!
- **Good UI:** Every car has the same pedal layout—drivers adapt easily.
- **Lesson:** Consistency in UI (icons, navigation, colors) helps users feel familiar and confident.

5. Feedback → Elevator Button Analogy

- **Bad UI:** You press the elevator button, but there's no light or sound—you don't know if it worked.
- **Good UI:** The button lights up or you hear a "ding," giving instant confirmation.
- **Lesson:** Feedback in UI tells users their action worked.

6. Simplicity → TV Remote Analogy

- **Bad UI:** A remote with 50 buttons, most unused by average users.
- **Good UI:** A remote with only essential buttons—power, volume, channel, menu.
- **Lesson:** Too many options confuse users; fewer, well-placed controls improve usability.

7. Accessibility → Building Design Analogy

- **Bad UI:** A building without ramps excludes wheelchair users.
- **Good UI:** A building with ramps, elevators, and braille signs is inclusive.
- **Lesson:** Accessible UI designs ensure everyone, including people with disabilities, can use the system.

Key points from the syllabus:

- Human behavior: memory, perception, problem-solving.
- Computer systems: devices, interaction models, ergonomics.
- Design principles: golden rules, cognitive models, usability engineering.
- Practical part: Game development using Unity + C#.

Importance of HCI

Area	How HCI Helps	Example
User-friendly Apps	Improves user experience & satisfaction	Amazon, Swiggy UI design
Productivity Tools	Reduces errors, makes tasks faster	MS Word, Google Docs
Safety-Critical Systems	Ensures safe operations	Medical devices, ATMs
Game Development	Enhances engagement & fun	Unity-based games
Emerging Tech	AR/VR, AI interfaces, Conversational systems	Alexa, ChatGPT, AR apps

“If technology is hard to use, people won’t use it—even if it’s powerful. HCI ensures technology serves humans effectively.”

“By the end, you will be able to design and evaluate user interfaces and implement interactive applications.”

1. ♦ Human Input/Output (I/O) Channels

Humans interact with computers through sensory and motor systems, which serve as input (perception) and output (action) channels.

► Visual Channel

- Dominant sense in most HCI tasks.
- Involves the eyes interpreting shapes, colors, motion, spatial layout.
- Critical in GUI design — icons, colors, typography, alignment.
- Design Considerations:
 - Avoid visual clutter.
 - Ensure contrast and readability.
 - Use visual hierarchy to guide user attention.

► Auditory Channel

- Involves hearing sounds such as alerts, speech, and background audio.
- Useful in environments where the user's visual attention is limited.
- Examples:
 - Error beeps.
 - Voice interfaces (e.g., Siri, Alexa).
 - Text-to-speech systems.
- Design Tip: Avoid using sound alone for critical information (accessibility).

► Haptic Channel

- Involves touch and proprioception (sense of body position).
- Examples:
 - Vibration feedback in phones.
 - Force feedback in game controllers or simulators.
 - Touch gestures on screens or trackpads.
- Enhances immersion and physical realism in interfaces.

◆ Human Memory

Memory is essential in interaction design since users must remember commands, locations, or sequences.

► Sensory Memory

- Registers immediate input from senses.
- Very short duration: ~200 ms (visual), ~2-4 seconds (auditory).
- Holds exact copy of sensory input.
- Example: Flash of lightning still "visible" in mind for a second.
- ● **Example:**
A flash of a pop-up ad on a website may disappear quickly, but your eyes briefly retain the image.
- ➡ **Why it matters:** Designers must ensure critical info stays long enough to register consciously.

► Short-Term Memory (STM)

- Also known as Working Memory.
- Temporarily holds information during cognitive tasks.
- Capacity: 7 ± 2 chunks of information.
- Retention: 15–30 seconds without rehearsal.
- Design Insight:
 - Avoid forcing users to memorize long steps.
 - Use recognition rather than recall (e.g., menus instead of commands).

● **Example:**

While booking a train ticket on IRCTC:

- You copy the OTP from SMS and paste it into the site.
- You can remember it temporarily (e.g., 456721) — fits the 7 ± 2 rule.

➡ **Why it matters:** Interfaces should reduce dependency on memory (e.g., copy-paste options, auto-fill).

► Long-Term Memory (LTM)

- Stores information for longer durations: hours to years.
- Practiced/repeated data move from STM to LTM.
- Two types:
 - Declarative (explicit): Facts, concepts.
 - Procedural (implicit): Skills and routines.

- **Design Implication:**

- Familiar icons and conventions reduce learning time.
- Training, consistency, and meaningful content help users retain knowledge.

- **Example:**

Regular users of MS Word remember:

- Ctrl + C = Copy
- Ctrl + Z = Undo

➔ *Why it matters:* Consistent shortcuts and interface elements become stored as skills over time.

◆ Reasoning and Problem Solving

Cognitive processes used when interacting with systems, solving errors, or learning new features.

► Deductive Reasoning

- Starts with a general rule → applies it to a specific situation → derives a conclusion.
- Example:
 - Rule: "All menu options are accessible under 'Settings'."
 - Task: "I need to change the language, so it must be under 'Settings'."

- **Example:**

In online forms, users know:

- "A red asterisk (*) means required field."
- They apply this rule to new forms.

➔ *Why it matters:* Use of standard design conventions supports deductive reasoning.

► Inductive Reasoning

- Observes patterns in specific examples → infers a general rule.
- Example:
 - "Every time I swipe down, the page refreshes → Swiping down = Refresh."

- **Example:**

In Instagram, every time users swipe down, the feed refreshes.

- They observe the pattern → generalize that "Swipe down = Refresh" on similar apps.

➔ *Why it matters:* Consistency across apps improves user learning through induction.

► Problem-Solving Strategies

1. Trial-and-Error

- User attempts various actions until something works.
- Common in poorly designed systems or when users are unfamiliar.
- Risk: Time-consuming or frustrating if feedback is poor.

● **Example:**

When adjusting settings on a new smart TV, a user may:

- Click random icons to figure out which one controls brightness.

➡ *Why it matters:* Clear icons, labels, and feedback reduce user frustration.

2. Heuristics

- Using simple rules or “mental shortcuts” to make decisions.
- Example: “If the app isn’t responding, close and reopen it.”
- Designers use heuristic evaluation to test interfaces.

● **Example:**

If an app freezes, the user:

- Closes and reopens it before contacting support.

➡ *Why it matters:* Interfaces should support common troubleshooting actions like “Refresh,” “Back,” or “Restart.”

3. Analogies

- Solving new problems based on similar past experiences.
- Example: If a web form behaves like a paper form, users apply the same logic.

● **Example:**

Online shopping cart mimics a real-world cart:

- Add items, review, remove, then checkout.

➡ *Why it matters:* Using metaphors helps users apply past knowledge to new systems.

4. Rule-Based Logic

- Applying known “if-then” rules to navigate or resolve issues.
- Example: “If I click the trash icon, then the file will be deleted.”

● **Example:**

In file managers (e.g., Windows Explorer):

- Right-click + Delete = file moves to Recycle Bin

➔ *Why it matters:* Predictable actions build user confidence and reduce errors.

💡 **Design Implications for HCI**

- Interfaces should support human memory limits (e.g., progressive disclosure, breadcrumbs).
- Provide clear feedback to guide problem-solving.
- Use multiple I/O channels (visual + auditory + haptic) for enhanced user experience.
- Design with users' reasoning processes in mind — intuitive layouts and consistent patterns reduce cognitive load.

2. The computer: Devices – Memory – Processing and Networks

◆ **Devices**

➤ **Input Devices**

These allow users to send information *into* the computer.

Device	Real-Time Example	Application
Keyboard	Typing a report in MS Word	Data entry, coding, writing
Mouse	Clicking icons on a desktop	GUI interaction
Touchpad	Navigating on a laptop	Mouse alternative
Scanner	Digitizing a paper document	Uploading signed forms
Sensors	Fingerprint scanner on phone	Biometric authentication, IoT data collection

➤ **Output Devices**

These allow the computer to communicate results to the user.

Device	Real-Time Example	Application
Monitor	Watching a video on YouTube	Displaying visuals, reading documents
Printer	Printing a bank statement	Hard copy documentation
Speakers	Listening to music or system alerts	Audio output, accessibility
VR Headset	Playing VR games	Immersive simulations, training

► **Combined Input/Output Devices**

Devices that can both receive and display information.

Device	Real-Time Example	Application
Touchscreen	Using a smartphone	Direct manipulation, kiosks
Stylus Pen Input	Writing notes on a tablet	Digital art, handwriting input
Voice Assistant	Talking to Alexa or Siri	Smart home control, voice interaction

◆ **Memory**

► **Primary Memory**

Used for temporary storage and fast access by the CPU.

Type	Real-Time Example	Purpose
RAM	Running multiple tabs in Chrome	Stores active applications and data
ROM	Booting your laptop	Contains BIOS – system startup code

🧠 *Primary memory is fast but volatile (RAM loses data when power is off).*

► **Secondary Memory**

Permanent storage of files and programs.

Type	Real-Time Example	Use Case
HDD (Hard Disk Drive)	Storing movies, games	Mass storage, slower but cheaper
SSD (Solid-State Drive)	Fast boot-up in laptops	Faster access, no moving parts

💡 *Most modern laptops use SSDs for speed and reliability.*

► **Cache Memory**

- High-speed memory between CPU and RAM.
- Stores frequently accessed data to reduce processing time.

● **Example:**

When typing a document, predictive text or auto-correct may feel instantaneous — thanks to cached data.

◆ Processing

► CPU (Central Processing Unit)

- Known as the "brain" of the computer.
- Executes instructions and controls operations.

● Example:

When using Excel, the CPU performs calculations instantly.

► GPU (Graphics Processing Unit)

- Specialized processor for rendering images, animations, and videos.

● Example:

Playing Fortnite or editing videos in Adobe Premiere Pro — GPU handles the visuals.

► Embedded Processors

- Microcontrollers inside IoT devices, smartphones, or appliances.

● Examples:

- Smartwatches: Track steps, heart rate (e.g., Apple Watch).
- Smart TVs: Run apps like Netflix.
- Washing Machines: Automatically adjust wash cycle based on load.

◆ Networks

Computer networks connect multiple systems to share data and resources.

► Types of Networks

- LAN (Local Area Network): Office or home Wi-Fi
- WAN (Wide Area Network): The Internet
- PAN (Personal Area Network): Bluetooth between phone and smartwatch

► Real-Time Examples

Network Application	Description
Cloud Storage (e.g., Google Drive)	Stores files online, access from anywhere
Video Conferencing (Zoom, MS Teams)	Real-time audio-video communication
Online Gaming (e.g., PUBG)	Requires fast response via low-latency networks
Remote Work Tools	Accessing desktop environments or servers from different locations

► **Impact on HCI**

- High latency = Poor user experience (e.g., video buffering)
- Unstable networks = Interruptions in UI interaction (e.g., form submission errors)

✓ **HCI Design**

Component	HCI Consideration
Input Devices	Choose based on user environment (e.g., touch in mobile)
Output Devices	Use accessible output (e.g., audio for visually impaired)
Memory	Optimize for memory limits (use caching, reduce reloads)
Processing	Use background processing to avoid UI freezing
Networks	Provide offline mode or save state in poor connections

3. Historical Evolution of HCI

◆ **Evolution Milestones in HCI**

The evolution of Human-Computer Interaction (HCI) reflects how technology shifted from machine-focused to human-centered design. Here's a chronological breakdown with real-world examples:

1. Batch Processing Era (1940s–1960s)

- Description: Computers operated without user interaction. Users submitted jobs (punched cards) and received output later.
- Interface: None; physical card input → delayed output.
- User Role: Passive — no real-time feedback.

● **Real-Time Example:**

Early mainframe computers like IBM 1401.

🧠 **HCI Limitation:**

No interaction; not user-friendly or immediate.

2. Command-Line Interface (CLI) (1960s–1980s)

- Description: Users typed commands into a text-based interface.
- Features: Requires syntax knowledge and memorization.

● **Real-Time Examples:**

- MS-DOS

- Unix Terminal
- Windows PowerShell (still in use)

 **Applications:**

Used by developers, system admins, and programmers.

 **HCI Note:** Powerful but not beginner-friendly; high learning curve.

3. Graphical User Interface (GUI) (1980s–2000s)

- Description: Introduced visual elements (windows, icons, menus, pointers).
- Interaction: Mouse-based, visual navigation.

 **Real-Time Examples:**

- Microsoft Windows
- macOS
- Desktop Linux (GNOME, KDE)

 **Applications:**

- Word processing (MS Word)
- Spreadsheets (Excel)
- Drag-and-drop file operations

 **HCI Note:** Major usability improvement; enabled mass computer adoption.

4. Web Interface (1990s–Present)

- Description: Interaction via browsers using hyperlinks, forms, and dynamic content.
- Access: From anywhere, platform-independent.

 **Real-Time Examples:**

- E-commerce (Amazon, Flipkart)
- Email (Gmail, Outlook)
- Learning platforms (Coursera, Google Classroom)

 **Applications:**

- Online shopping
- Banking
- Social media
- Content management systems (CMS)

✦ **HCI Note:** Introduced hypertext, navigation, multimedia, and global access.

5. Mobile & Touch Interfaces (2007–Present)

- Description: Enabled direct manipulation via fingers and gestures.
- Input: Touch, swipe, pinch, voice.

● **Real-Time Examples:**

- Smartphones (Android, iOS)
- Tablets (iPad)
- Touch-enabled kiosks

🧠 **Applications:**

- Mobile apps (WhatsApp, Swiggy)
- Voice assistants (Google Assistant, Siri)
- Touch ATMs, self-check-in kiosks

✦ **HCI Note:** Made interfaces intuitive and portable; users expect instant and personalized interactions.

6. Natural User Interfaces (NUI) – Present and Emerging

- Description: Allows interaction through natural means — gesture, voice, eye movement, and augmented reality.
- Tech: AR/VR, haptics, brain-computer interfaces.

● **Real-Time Examples:**

- AR: Pokémon GO, IKEA Place app
- VR: Oculus Quest, virtual training simulators
- Haptics: Wearable fitness bands with vibration
- Voice: Amazon Alexa, smart TVs

🧠 **Applications:**

- Immersive learning
- Virtual tours
- Smart homes
- Healthcare assistive tech (e.g., eye-controlled wheelchairs)

✦ **HCI Note:** Fosters immersive, context-aware, and touchless interfaces.

◆ Key Drivers in HCI Evolution

▶ 1. Advances in Hardware

- Faster CPUs, affordable memory, better displays, sensors, touchscreens, AR/VR gear.
- Examples:
 - Evolution from CRT to LCD to foldable screens.
 - Multi-core processors enabling smooth multitasking.

 **Impact:** Enables richer, faster, and more responsive interactions.

▶ 2. Cognitive Science Insights

- Understanding how humans perceive, learn, and decide.
- Led to:
 - Recognition > Recall design
 - Fitts's Law in UI spacing
 - Consistency, feedback loops

 **Example:**

Designing ATM interfaces with clear language and icons based on user mental models.

 **Impact:** Interfaces became intuitive, reducing cognitive load and user errors.

▶ 3. Increased User Expectations

- As users got familiar with smartphones, apps, and AI, expectations rose for:
 - Personalization
 - Instant response
 - Voice and gesture control
 - Aesthetically pleasing UI

 **Examples:**

- Instant loading in apps like Instagram.
- Auto-suggestions in Google Search.
- Dark mode, minimal design in modern UIs.

 **Impact:** HCI became more user-centered, focusing on experience (UX) over just functionality.

✓ HCI Evolution Milestones

Era	Interface Type	Key Feature	Real-World Example
1950s–60s	Batch Processing	No interaction	IBM Mainframe
1970s–80s	CLI	Text-based commands	MS-DOS, Unix
1980s–2000s	GUI	Windows, Icons, Mouse	Windows, macOS
1990s–Today	Web Interface	Hyperlinks, Browsers	Amazon, Gmail
2007–Today	Touch Interface	Swipe, Tap, Voice	iPhone, Android
2010s–Future	NUI	Gesture, AR/VR, AI	Oculus, Alexa

4. Interaction in HCI

◆ Interaction Models

Interaction models describe how users interact with systems, helping designers predict and improve usability.

➤ 1. Norman's Execution-Evaluation Cycle

Describes interaction as a two-phase loop:

- Execution: User forms goal → plans action → executes.
- Evaluation: System gives feedback → user interprets → adjusts behavior.

● *Real-Time Example:*

Using an ATM:

1. Goal: Withdraw ₹1000
2. Plan: Insert card → choose withdraw → enter amount
3. Execute: Press buttons
4. Feedback: Message confirms transaction → cash dispensed
5. Evaluate: User checks amount received and balance

🧠 **Application:** Helps identify where users get stuck — improves design of step-by-step interfaces.

➤ 2. Abowd and Beale's Interaction Framework

Components:

User → Input → System → Output → Feedback → User

● *Real-Time Example:*

In a drawing app:

- User draws using a stylus.

- Input: Touch/stylus location.
- System: Updates canvas.
- Output: Displays updated drawing.
- Feedback: Visual change confirms success.

 **Application:** Helps designers model and debug interaction flow.

◆ Interaction Frameworks

Used to organize the system's internal structure and separate concerns for easier design and development.

► 1. Seeheim Model

- Divides system into 3 parts:
 - Presentation (UI),
 - Dialogue Controller (interaction logic),
 - Functional Core (underlying logic)

● **Example:** Traditional software like tax filing systems, where UI is separate from logic and data.

► 2. MVC (Model-View-Controller)

- Model: Manages data
- View: Displays data
- Controller: Handles user input

● **Real-Time Examples:**

- Web apps (React.js, Angular)
- **Instagram:**
 - Model: User profile data
 - View: Profile screen
 - Controller: Like/comment button actions

 **Application:** Encourages modular development and easier updates.

◆ Ergonomics in HCI

Study of optimizing physical interactions between humans and devices.

► Key Considerations:

- Posture and reach
- Input device design
- Screen angle and brightness
- Accessibility (for elderly, differently-abled)

● Real-Time Examples:

- Ergonomic keyboards and vertical mice to prevent wrist strain.
- Adjustable standing desks.
- Mobile UI designed for one-hand thumb use.

 **Application:** Prevents fatigue and injuries, enhances comfort and performance.

◆ Interaction Styles

How users communicate with systems.

Style	Description	Real-Time Example
Command Line	Text input commands	Linux Terminal, Git
Menu-Based	Choose from list	ATM options, Settings menus
Form Fill-In	Input in fields	Online registration, bank forms
Direct Manipulation	Interact with objects visually	Drag-and-drop in file explorer
Natural Language	Use voice/text	Chatbots (e.g., ChatGPT), Alexa

 **Application:** Choosing the right style improves usability and user satisfaction.

◆ Elements of Interaction

The building blocks of user interfaces.

Element	Role	Example
Input Controls	Allow user action	Buttons, checkboxes, sliders
Output Displays	Show system state	Labels, graphs, messages
Feedback Mechanisms	Communicate result	Progress bars, alerts, sound

 **Real-Time Example:**

- E-commerce checkout:
 - Button to “Place Order”
 - Spinner shows processing
 - Confirmation message

 **Interactivity**

Measures how dynamic the interaction is between user and system.

▶ **Factors:**

- Responsiveness: Delay between input and system response.
- Control: How much influence the user has over outcomes.
- Dialog flow: Quality of interaction, smoothness.

 **Real-Time Examples:**

- High Interactivity: Video games, real-time chat, drag-and-drop UIs.
- Low Interactivity: Static web pages, info kiosks with limited options.

 **Application:** Critical for designing engaging and responsive systems (e.g., search-as-you-type in Google).

 **Paradigms in HCI**

Paradigms are conceptual models or trends that define how interaction is approached.

▶ **1. WIMP (Windows, Icons, Menus, Pointer)**

- Traditional desktop metaphor
- Used in GUIs (Windows, macOS)

 **Real-Time Example:**

Using MS Word — pointer clicks icons and menus to format text.

▶ **2. Ubiquitous Computing**

- Computing embedded in everyday objects
- Invisible interaction

 **Real-Time Example:**

- Smart home devices: Lights turn on when someone enters.

- RFID-based attendance systems

▶ 3. Context-Aware Systems

- Adapt based on user's context: location, time, activity

● **Real-Time Example:**

- Google Maps suggesting commute routes during peak traffic.
- Smartwatch muting notifications during meetings (calendar sync)

▶ 4. Augmented & Virtual Reality

- AR overlays digital content on the real world.
- VR immerses the user in a completely digital environment.

● **Real-Time Examples:**

- AR: IKEA app for placing furniture virtually.
- VR: Oculus Rift for immersive gaming or architecture walkthroughs.

 **Application:** Used in education, design, training, healthcare, and entertainment.

✔ **Interaction Components and Examples**

Component	Real-World Application
Interaction Models	ATM workflow, drawing apps
Frameworks	Web and mobile app architecture
Ergonomics	Standing desks, one-hand mobile design
Styles	Voice assistants, drag-drop editors
Elements	Submit button, alert boxes
Interactivity	Google Search auto-suggest
Paradigms	Smart homes, VR training, AR shopping