

Teaching Accessibility

Fundamental Concepts & Skills

The members of Teach Access have identified a set of Fundamental Concepts and Skills necessary to understand and implement inclusive design and development of technology for people with diverse abilities, including people with disabilities.

The document was prepared at the request of numerous faculty, to guide the infusion of these topics into mainstream courses in design, computer science, engineering, human-computer interaction, user experience, industrial design, usability, ergonomics and related fields.

Resources for teaching these materials are widely available, and a curated collection will be posted in the [Teach Access Resource Center](#).

- **Understanding Disability**
 - Common types of disabilities & current demographics - understanding ability from a **functional** approach and the impact of disability in daily living activities
 - Physical and Sensory
 - Vision (Blindness & Low Vision)
 - Hearing
 - Motor
 - Speech
 - Physical
 - Cognitive
 - Memory
 - Problem-Solving
 - Attention
 - Thinking
 - Reading, Linguistic and Verbal Comprehension
 - Math Comprehension
 - Visual Comprehension
 - Concepts
 - Temporary, Permanent, Situational limitations
 - Spectrum of ability
 - Experiences
 - Videos of and meetings with self-advocates and family members
 - Video Conferencing with representatives from advocacy organizations
 - Videos of engineers and computer scientists who have worked on accessibility, relating their experience.

- **Societal Context & Historical Perspective**

- Identity & Cultural Norms
- Establishment of regulatory requirements & standards
 - Accessibility as a civil right
 - ADA history & basic rules
 - W3C and Web Content Accessibility Guidelines (WCAG)
 - Section 508 history & basic rules
 - 21st Century Communications and Video Accessibility Act (CVAA) legislation and FCC regulations
 - Landmark legal cases
 - Pending regulations and implementations (EU, UNCRPD)
- **User Interface Facilitators & Barriers (w/ examples)**
 - Most technologies and interaction models require accessibility design thinking
 - Usability + Accessibility = user delight
 - GUIs
 - Facilitator: Screen reader support
 - Barrier: No screen reader support
 - Touch/Tactile Screens
 - Facilitator: Big touch targets
 - Barrier: Small touch targets
 - Gesture Input
 - Facilitator: Intuitive interaction model for performing a task (pinch to zoom)
 - Barrier: No alternative input mechanism for people with limited dexterity
 - Video & Rich Media
 - Facilitator: Rich storytelling medium
 - Barrier: No closed captions
 - Speech to Text / Speech Commands
 - Facilitator: Ease of text input and hands-free interaction
 - Barrier: Need alternative interaction model for speech and audio I/O
 - Help & Support Considerations
 - Providing accessible instructions and support
 - Training on AT and accessibility features
 - Experiences
 - Testimonials from people with disabilities
 - Simulation tools/techniques for design and testing
- **Common Assistive Technologies (AT)**
 - Types (screen reader, captioning, switches, magnifiers, braille display, etc.)
 - Evolution from specialty AT to mainstream AT (Integrated into TVs, PCs, smartphones, tablets, OTT boxes, etc.)
 - Profound impact of mainstream technologies (autonomous cars, speech recognition, language translation tools)

- Simple tips to get students introduced to using AT.
- Experiences
 - Demonstration of a screen reader (or alternative AT)
 - Simulation of a screen reader (one person uses keyboard, partner can see screen and acts as the "screen reader")
 - Hands-on with a screen reader (or alternative AT)
 - Partner with AT providers or Assistive Technology Act Programs and observe an AT assessment
- **Best Practices for Product Development**
 - Accessibility is a design principle, not a feature set
 - Accessible Design as part of Universal Design
 - User Research & Design Principles
 - Requirements gathering
 - Designing and conducting inclusive user studies
 - User feedback (beta, post-launch, etc)
 - Universal design (global audience) incorporating personalization (considerations of the individual)
 - Development Cycle Integration
 - Leadership
 - Product Manager
 - Design & Research
 - Engineering & implementation
 - QA
 - Validation and user testing (ie, testing in screen readers & zoom)
 - Communication out (ie, documentation, marketing and customer support)
 - Communicating up (ie, ROI, impact analysis)
 - Experiences
 - Examples from companies/work groups on how they've integrated accessibility into their processes
 - Empathy exercise around a challenging interaction
- **Applied Techniques**
 - Computer Science
 - Coding best practices
 - UI elements & properties (label, focus, tab order)
 - Semantic code (headings, landmarks, buttons)
 - Accessibility APIs and frameworks (iOS, Android, ARIA)
 - Languages and platforms that support accessibility
 - Examples: HTML, JS, ARIA, iOS, Android
 - Testing tools
 - Experiences

- Make an existing app more accessible and then create the same app with accessibility support built in from the beginning
 - Require any app project to include accessibility support and testing verification
 - Make an app that can only be used with assistive technology (no visible labels so a screen reader must be used; controls don't respond to mouse input so keyboard shortcuts have to be used; really tiny writing so a screen magnifier has to be used...)
 - Create a module/app which uses non-visual I/O to interact with complex 'visual' elements such as scatter plots and infographics
 - Compare a 'gold standard' product with a failed product
- Design Best Practices
 - Visual & Interaction Design
 - Color contrast
 - Color choice and meaning
 - Tap targets
 - Gestures and alternative inputs
 - UI and interactive design
 - [POUR](#) (Perceivable, Operable, Understandable, Robust)
 - Usability
 - User research (e.g., interviews, personae, user testing)
 - Use cases
 - Information architecture
- Universal vs. Individual
 - Optimize for diverse user base
 - Allow user customization where appropriate (i.e., font size)
- Design tips
 - Avoid flickering content
 - Avoid complex language
 - Avoid unrecoverable errors
 - Avoid timed content
 - Avoid turning off/circumventing default interactions
- Design Tools
 - Color contrast analysers
 - Reference documentation (see [Teach Access Resource Center](#))
- Experiences
 - Make an existing app more accessible
 - Include people with disabilities in user testing
 - Require any app project to include accessibility support and testing verification