**English Course Summary - BINV2120-1**

**Leonardo da Vinci University-College**  
**Academic Year 2023-2024**

**Table of Contents**

1. [Public Speaking](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#1-public-speaking)
2. [Preparing Your First CV](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#2-preparing-your-first-cv)
3. [Writing Professional Emails](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#3-writing-professional-emails)
4. [Job Interviews](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#4-job-interviews)
5. [Dark Patterns in Digital Design](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#5-dark-patterns-in-digital-design)
6. [Mobile Applications: Native, Web & Hybrid](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#6-mobile-applications-native-web--hybrid)
7. [Introduction to Programming Languages](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#7-introduction-to-programming-languages)
8. [Programming Languages Deep Dive](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#8-programming-languages-deep-dive)
9. UI/UX
10. [Ergonomics in the Workplace](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#9-ergonomics-in-the-workplace)
11. [Web Tracking and Privacy](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#10-web-tracking-and-privacy)
12. [Mock Exam](https://claude.ai/chat/0a72df5a-a5b6-4c8e-9757-d4f7e85c1a08#11-mock-exam)

**1. Public Speaking**

**Key Concepts**

Public speaking is the art of communicating effectively to an audience. It involves careful preparation, confident delivery, and engaging presentation techniques.

**Do's and Don'ts**

**✔️ Do's:**

* **Prepare Thoroughly**: Research your topic, organize thoughts, create well-structured content
* **Know Your Audience**: Understand needs, interests, and expectations
* **Start Strong**: Begin with compelling openings (story, fact, quote, rhetorical question)
* **Be Clear and Concise**: Use simple, straightforward language
* **Use Visual Aids Wisely**: Enhance message rather than distract
* **Engage with Eye Contact**: Establish connection and convey confidence
* **Speak Clearly and Slowly**: Articulate words clearly, maintain moderate pace
* **Vary Your Voice**: Use tone, pitch, volume for emphasis
* **Use Body Language**: Gesture naturally, maintain confident posture
* **Tell Stories**: Make messages relatable and memorable

**❌ Don'ts:**

* **Don't Overload with Information**: Focus on key points, keep concise
* **Don't Read Directly**: Use notes as reference, not script
* **Don't Use Filler Words**: Minimize "um," "uh," "like," "you know"
* **Don't Turn Your Back**: Always face audience
* **Don't Rush**: Practice pacing for clarity
* **Don't Wing It**: Thorough preparation is essential
* **Don't Use Inappropriate Humor**: Avoid offensive or alienating content

**Key Phrasal Verbs**

* **Bring up**: Mention or introduce a topic
* **Break down**: Simplify complex information
* **Go over**: Review content
* **Run through**: Practice or rehearse
* **Speak up**: Talk louder
* **Hold on**: Wait a moment
* **Point out**: Highlight or emphasize
* **Start off**: Begin
* **Wrap up**: Conclude
* **Turn to**: Move to a different topic

**Overcoming Fear of Public Speaking**

1. **Understand Your Fear**: Acknowledge nervousness is normal
2. **Practice**: Rehearse multiple times
3. **Know Your Material**: Become expert on topic
4. **Visualize Success**: Imagine positive outcomes
5. **Control Environment**: Arrive early, test equipment
6. **Practice Relaxation**: Use breathing exercises, meditation
7. **Use Positive Self-Talk**: Replace negative thoughts with affirmations
8. **Start Small**: Begin with smaller audiences
9. **Engage Audience**: Make connections through interaction
10. **Focus on Message**: Concentrate on value rather than anxiety

**2. Preparing Your First CV**

**CV Structure and Best Practices**

**Essential Components:**

1. **Contact Information**: Full name, phone, email, LinkedIn profile
2. **Professional Summary**: Brief career goals statement
3. **Professional Experience**: Reverse chronological order with achievements
4. **Education**: Most recent degree first
5. **Skills**: Technical and soft skills relevant to job
6. **Additional Sections**: Certifications, memberships, publications (optional)

**Do's and Don'ts**

| **CV Do's** | **CV Don'ts** |
| --- | --- |
| ✔️ Include comprehensive work history | ❌ Don't include irrelevant details |
| ✔️ Use clear and concise language | ❌ Avoid jargon or acronyms |
| ✔️ Use professional organized format | ❌ Don't use fancy fonts or colors |
| ✔️ Customize for each job application | ❌ Don't submit generic CV |
| ✔️ Highlight specific accomplishments | ❌ Don't provide vague descriptions |
| ✔️ Include relevant industry keywords | ❌ Don't overstuff with keywords |
| ✔️ Thoroughly proofread for errors | ❌ Don't rely solely on spellcheck |
| ✔️ Tailor length based on experience | ❌ Don't make excessively long |

**Key Phrasal Verbs for CVs**

* **Kick-off**: Start or begin
* **Line up**: Align with goals
* **Come up with**: Generate ideas
* **Put on**: Organize events
* **Wrap up**: Complete projects
* **Bring in**: Generate revenue
* **Put forward**: Suggest ideas
* **Set up**: Establish partnerships
* **Lead to**: Result in outcomes
* **Put together**: Assemble teams
* **Take on**: Accept leadership
* **Bring about**: Foster development
* **Thrive on**: Excel at challenges
* **Deliver**: Achieve results

**Cover Letter vs CV**

| **Cover Letter** | **CV** |
| --- | --- |
| Brief, personalized document | Comprehensive work history overview |
| Introduces and explains interest | Presents detailed qualifications |
| Highlights relevant skills/experiences | Lists education, experience, skills |
| Usually one page | Can be multiple pages |
| Addressed to hiring manager | Shared with employers/recruiters |
| Persuasive tone | Factual presentation |

**3. Writing Professional Emails**

**Email Etiquette Guidelines**

**✔️ Do's:**

* **Use Professional Email Address**: Reflect name or professional identity
* **Clear Subject Line**: Accurately summarize email content
* **Appropriate Greeting**: Match formality to relationship
* **Concise Message**: Be clear and to the point
* **Proper Grammar/Spelling**: Proofread before sending
* **Mindful Tone**: Use polite, respectful language
* **Smart Formatting**: Use paragraphs, bullet points appropriately
* **Prompt Response**: Reply in timely manner
* **Thoughtful Cc/Bcc**: Include only necessary recipients
* **Professional Signature**: Include contact information

**❌ Don'ts:**

* **Inappropriate Email Address**: Avoid unprofessional addresses
* **Empty Subject Line**: May be overlooked or marked as spam
* **All Caps**: Appears as shouting
* **Excessive Exclamation Points**: Seems overly emotional
* **Chain Emails**: Don't forward spam or irrelevant content
* **Unnecessary Reply All**: Clutters inboxes
* **Excessive Slang**: Maintain professional tone
* **Large Attachments**: Warn recipients or use file-sharing
* **Sensitive Topics**: Use secure channels for private matters
* **Missing Attachments**: Check before sending

**Common Email Phrasal Verbs**

* **Push back**: Postpone or delay
* **Put off**: Postpone
* **Follow up**: Check status or progress
* **Figure out**: Find solution
* **Go over**: Review or discuss
* **Set up**: Establish or arrange
* **Check out**: Verify or examine
* **Sort out**: Resolve problems
* **Call off**: Cancel
* **Look into**: Investigate

**Email Components Template**

Subject: Clear, descriptive subject line

Dear [Name]/Hello [Team],

Opening paragraph: State purpose clearly

Middle paragraph(s): Provide details, context, requests

Closing paragraph: Next steps, thanks, call to action

Best regards/Sincerely,

[Your name]

[Contact information]

**4. Job Interviews**

**Interview Preparation**

**✔️ Do's:**

* **Research Company**: Learn mission, values, culture, products/services
* **Review Job Description**: Understand requirements and responsibilities
* **Practice Common Questions**: Prepare answers for typical questions
* **Prepare Questions**: Develop thoughtful questions about role/company
* **Dress Professionally**: Align with company culture
* **Arrive Early**: 10-15 minutes before scheduled time
* **Bring Resume Copies**: Have extras available
* **Use STAR Method**: Structure behavioral question responses
* **Show Enthusiasm**: Express genuine interest
* **Listen Actively**: Pay attention to questions and statements

**❌ Don'ts:**

* **Don't Arrive Late**: Punctuality is crucial
* **Don't Overlook Appearance**: Dress appropriately
* **Don't Speak Negatively**: Avoid criticizing past employers
* **Don't Interrupt**: Let interviewer finish questions
* **Don't Exaggerate/Lie**: Be honest about qualifications
* **Don't Be Unprepared**: Lack of preparation shows disinterest
* **Don't Forget Follow-up**: Send thank-you note after interview

**The STAR Method**

**S**ituation: Set the context and background   
**T**ask: Describe your responsibility or objective  
**A**ction: Explain steps you took to address situation   
**R**esult: Share outcomes and impact of your actions

**Example STAR Response:**

**Question**: "Can you give me an example of a time when you had to resolve a conflict within your team?"

**Situation**: While working as project manager at XYZ Company, two team members had differing opinions on project approach for critical delivery.

**Task**: Resolve conflict, ensure project stayed on track, maintain positive working relationships.

**Action**:

* Met separately with both team members to understand perspectives
* Identified common goals between their ideas
* Facilitated team meeting for open communication
* Mediated to find compromise incorporating both approaches' strengths

**Result**: Team completed project ahead of schedule and under budget. Conflict resolution improved communication for future projects.

**Key Interview Phrasal Verbs**

* **Bring in**: Introduce new ideas/revenue
* **Take on**: Accept responsibility
* **Go over**: Review experience/technologies
* **Elaborate on**: Provide more details
* **Carry out**: Execute tasks
* **Work on**: Focus on projects
* **Keep up with**: Stay current with trends
* **Dive into**: Explore deeply
* **Run into**: Encounter challenges
* **Figure out**: Find solutions

**5. Dark Patterns in Digital Design**

**Definition**

Dark patterns are user interface designs that are crafted to trick users into doing things they might not want to do, such as buying insurance with their purchase or signing up for recurring bills.

**Common Dark Pattern Types**

| **Dark Pattern** | **Definition** | **Example** |
| --- | --- | --- |
| **Forced Continuity** | User enrolled in ongoing cost service without clear consent | Free trial automatically converts to paid subscription |
| **Disguised Ads** | Ads designed to look like native content | Sponsored content resembling regular articles |
| **Friend Spam** | Users misled into sending invitations without clear consent | Social media auto-sending friend requests to entire contact list |
| **Hidden Costs** | Additional fees not transparently disclosed | Concert tickets with hidden service charges at checkout |
| **Misdirection** | Design guides users toward particular action while diverting attention | Bright "free trial" button hiding terms and conditions |
| **Cost Comparison Prevention** | Difficult to compare costs of different options | Travel site not displaying all fees together |
| **Privacy Zuckering** | Users encouraged to share more information than intended | Social media prompts for excessive personal information |
| **Trick Questions** | Questions designed to confuse users | "Opt-out of amazing offers" to trick into subscribing |
| **Roach Motel** | Easy to get into situation but difficult to leave | Simple subscription but complex cancellation process |
| **Sneak into Basket** | Items added to basket without clear consent | Subscription service automatically added to cart |
| **Confirm Shaming** | Users guilt-tripped into accepting offer | "No, I prefer to remain uninformed" decline option |
| **Bait and Switch** | Users enticed with one offer but presented with different option | Advertised discount leads to out-of-stock, expensive alternative |

**Ethical Implications**

* Manipulate user behavior without consent
* Exploit psychological vulnerabilities
* Prioritize company profits over user wellbeing
* Damage trust between users and digital platforms
* Create negative user experiences

**6. Mobile Applications: Native, Web & Hybrid**

**Types of Mobile Applications**

**Native Apps**

**Definition**: Applications developed specifically for particular operating system using platform-specific programming languages.

**Characteristics**:

* Written in platform-native languages (Swift/Objective-C for iOS, Java/Kotlin for Android)
* Optimal performance and responsiveness
* Full access to device hardware and OS features
* Platform-specific UI guidelines

**Pros**:

* Extensive device hardware utilization
* Highly performant and reliable
* Optimized user experience
* Future-proof investment

**Cons**:

* High upfront development cost
* Separate codebases for each platform
* Need different developers for each platform
* Longer time-to-market
* Updates require user download/reinstall

**Web Apps**

**Definition**: Applications accessed through web browsers, responsive to different screen sizes.

**Characteristics**:

* Built with HTML5, CSS, JavaScript
* Platform-independent
* Accessed via URL
* No installation required

**Pros**:

* Broad customer base across geographies
* Single version improves SEO
* Lower development costs
* No device storage required
* No app store downloads/updates needed

**Cons**:

* Browser-dependent functionality
* Can't work completely offline
* Limited device hardware integration
* Can't be sold through app stores

**Hybrid Apps**

**Definition**: Blend of native and web applications, using web technologies wrapped in native container.

**Characteristics**:

* Code written in HTML5, CSS, JavaScript
* Wrapped using plugins like Apache Cordova
* Downloaded from app stores
* Run inside native app using embedded browsers

**Pros**:

* Quicker and cost-effective development
* Single codebase for multiple platforms
* Good for MVP development
* Fast loading for content-heavy apps
* Plugin system accesses platform features

**Cons**:

* Lack power and speed of native apps
* Challenging UX and navigation patterns
* Higher load times for various elements
* Can't access all device features

**Cross-Platform Apps**

**Definition**: Applications that run on multiple platforms with shared codebase.

**Characteristics**:

* Broader scope than hybrid apps
* Built with technologies like React Native, Xamarin
* Mid-point solution between native and hybrid

**Pros**:

* Faster development than native
* Customizable for multiple platforms
* Satisfactory performance at affordable cost
* Good for games
* Native-like app experience possible

**Cons**:

* Longer debugging time
* Some platform-specific development still needed
* Complex projects increase time and cost
* Cross-platform security threats

**Key Mobile App Vocabulary**

| **Term** | **Definition** |
| --- | --- |
| **API** | Set of rules allowing software applications to communicate |
| **Native App** | Software designed specifically for particular mobile OS |
| **Web App** | Application accessed through web browser |
| **Cross-Platform Development** | Single codebase for multiple platforms |
| **User Experience (UX)** | Quality of user's interaction with application |
| **App Store** | Platform-specific digital distribution service |
| **Hybrid App** | Web technologies wrapped in native container |
| **Responsive Design** | Websites adapt to different screen sizes |
| **URL** | Address to access internet resources |
| **Mobile App** | Standalone software for mobile devices |

**Common Mobile App Phrasal Verbs**

* **Log in/out**: Access/exit account
* **Sign up for**: Register for service
* **Check out**: Examine or review
* **Turn on/off**: Enable/disable features
* **Set up**: Configure or establish
* **Sync up**: Synchronize data
* **Scroll through**: Navigate content
* **Fill out**: Complete forms
* **Back up**: Create data copies
* **Opt in/out**: Choose to participate or decline
* **Download**: Transfer files to device
* **Update**: Install latest version

**7. Introduction to Programming Languages**

**Popular Programming Languages Overview**

| **Language** | **Creator/Origin** | **Primary Use** | **Key Features** |
| --- | --- | --- | --- |
| **Python** | Guido van Rossum | Data science, web development, automation | Readability, simplicity |
| **JavaScript** | - | Web development | Language of the web |
| **Java** | Sun Microsystems | Enterprise software, Android apps | "Write once, run anywhere" |
| **C++** | Bjarne Stroustrup | Game development, system programming | High performance, memory control |
| **Swift** | Apple | iOS/macOS development | Speed, safety, Apple ecosystem |
| **PHP** | - | Web development, server-side scripting | Web scripting, easy integration |
| **Go** | Google | Cloud services, system programming | Concurrency support, simplicity |
| **Rust** | Mozilla | System programming, game engines | Memory safety, zero-cost abstractions |
| **SQL** | IBM | Database management | Querying relational databases |
| **Ruby** | - | Web development, scripting | Elegant syntax, dynamic typing |

**Java vs JavaScript**

| **Aspect** | **Java** | **JavaScript** |
| --- | --- | --- |
| **Type System** | Statically-typed, compiled | Dynamically-typed, interpreted |
| **Platform** | Server-side, mobile, desktop | Primarily web development |
| **Compilation** | Compiled to bytecode, runs on JVM | Interpreted by browsers/Node.js |
| **Syntax** | Similar to C++/C#, requires semicolons | Lightweight, influenced by C and Java |
| **Object Model** | Class-based inheritance | Prototype-based inheritance |
| **Ecosystem** | Large, mature enterprise ecosystem | Rich web development ecosystem |

**Programming Paradigms**

**Object-Oriented Programming (OOP)**

* Uses objects (instances of classes)
* Encapsulates data and behavior
* Promotes inheritance and polymorphism

**Multi-Paradigm**

* Supports multiple programming approaches
* Allows procedural, object-oriented, and functional programming
* Flexibility in problem-solving

**Compiled Languages**

* Source code translated to machine code before execution
* Examples: C++, Java (to bytecode)

**Dynamic Typing**

* Variable types determined at runtime
* Examples: Python, Ruby, JavaScript

**Platform Independence**

* Code runs on multiple platforms without modification
* Java achieves this through JVM

**Key Programming Concepts**

**Algorithm**: Step-by-step procedure for solving problems   
**Variable**: Storage location with associated symbolic name   
**Function**: Reusable block of code performing specific task   
**API**: Set of protocols for building software applications   
**Framework**: Pre-built software architecture providing foundation   
**Library**: Collection of pre-written code for specific tasks   
**Debugging**: Process of finding and fixing errors   
**Compilation**: Translating source code to machine code

**Programming Phrasal Verbs**

* **Look into**: Investigate problems
* **Figure out**: Solve algorithms
* **Work on**: Develop features
* **Run into**: Encounter bugs
* **Set up**: Configure development environment
* **Carry out**: Execute testing
* **Bring up**: Mention concerns
* **Log in**: Access accounts
* **Coming up with**: Creating algorithms
* **Take over**: Assume project maintenance

**8. Programming Languages Deep Dive**

**Programming Language Characteristics**

| **Language** | **Paradigm** | **Key Features** | **Common Use Cases** |
| --- | --- | --- | --- |
| **Python** | Multi-paradigm | Readability, extensive libraries | Web development, data science, automation |
| **Java** | Object-oriented | Platform independence (JVM), strong typing | Enterprise software, Android apps |
| **JavaScript** | Multi-paradigm | Front-end web scripting, asynchronous | Web development, browser automation |
| **C++** | Multi-paradigm | High performance, memory control | Game development, system programming |
| **Ruby** | Object-oriented | Elegant syntax, dynamic typing | Web development, scripting |
| **Swift** | Multi-paradigm | Speed, safety, Apple ecosystem | iOS/macOS app development |
| **Go (Golang)** | Compiled | Concurrency support, simplicity | Cloud services, system programming |
| **Rust** | Systems | Memory safety, zero-cost abstractions | System programming, game engines |
| **PHP** | Server-side | Web scripting, easy integration | Web development, server-side scripting |
| **SQL** | Declarative | Querying relational databases | Database management |

**Programming Paradigm Definitions**

**Object-Oriented**: Programming paradigm using objects (instances of classes) to represent and manipulate data and behavior. Focuses on organizing code into reusable, modular units with inheritance and encapsulation.

**Multi-Paradigm**: Programming languages supporting multiple paradigms (procedural, object-oriented, functional). Developers can choose the most suitable approach for their problem-solving needs.

**Compiled**: Languages requiring separate compilation step before execution. Source code is translated into machine code or intermediate code by compiler.

**Front-End**: Programming involving user interfaces and interactions on user's side, typically in web development.

**High Performance**: Languages designed to execute code efficiently and quickly, often offering low-level memory control for speed optimization.

**Dynamic Typing**: Type system where variable types are determined at runtime, as opposed to static typing where types are checked at compile-time.

**Platform Independence**: Code written in language can run on multiple platforms without modification.

**Concurrency Support**: Programming language capability allowing developers to write code performing multiple tasks concurrently or in parallel.

**Memory Safety**: Programming language property preventing common memory-related errors like null pointer dereferences and buffer overflows.

**Declarative**: Programming style where developers describe what they want to achieve rather than specifying how to achieve it step by step.

**Frameworks vs Libraries**

**Framework**

* **Definition**: Pre-built, comprehensive software architecture providing foundation for building applications
* **Characteristics**:
  + More opinionated, imposes specific design patterns
  + Fixed architecture that must be followed
  + Provides range of built-in functionalities, tools, APIs
  + Developers build within framework's constraints
* **Examples**: Ruby on Rails, Django (Python), Angular (JavaScript/TypeScript)

**Library**

* **Definition**: Collection of pre-written code/functions for performing specific tasks
* **Characteristics**:
  + More focused and specific functionality
  + Not opinionated about overall application structure
  + Modular integration - pick and choose components
  + No strict structure or flow imposed
* **Examples**: jQuery (JavaScript), NumPy (Python), Alamofire (iOS), JavaFX (Java)

**Advanced Programming Vocabulary**

* **Polymorphism**: Ability to process objects differently based on their data type or class
* **Encapsulation**: Bundling data and methods that work on that data within one unit
* **Inheritance**: Mechanism allowing new class to inherit properties and methods from existing class
* **Abstraction**: Hiding complex implementation details while showing only essential features
* **Concurrency**: Executing multiple tasks simultaneously
* **Asynchronous**: Programming allowing multiple operations to happen concurrently
* **Dependency Injection**: Design pattern providing dependencies to object rather than having object create them
* **Regression Testing**: Re-running functional tests to ensure previously working functionality still works
* **Microservices**: Architectural approach structuring application as collection of loosely coupled services

**9. User Experience & User Interface (UX/UI)**

**Introduction to UX and UI**

**Definitions and Core Differences**

| **Aspect** | **User Experience (UX)** | **User Interface (UI)** |
| --- | --- | --- |
| **Definition** | Focuses on the overall experience and satisfaction of the user with a product or service | Deals with the visual and interactive elements of a product, emphasizing its look and feel |
| **Scope** | Encompasses the entire user journey, including emotions, perceptions, and interactions | Primarily concerned with the visual aspects, such as layout, colors, typography, and interactive elements |
| **Goal** | Aims to create a positive, meaningful, and seamless experience for the user | Aims to provide an aesthetically pleasing and user-friendly interface |
| **Components** | Includes usability, accessibility, information architecture, and overall satisfaction | Encompasses visual design, layout, responsiveness, and interactive elements |
| **Research** | Involves user research, testing, and analysis to understand user needs and behaviors | May involve usability testing, but primarily focuses on visual and interactive design principles |
| **Emphasis** | Places emphasis on the user's emotions, perceptions, and the overall journey | Places emphasis on the visual appeal, accessibility, and usability of the interface |
| **Iterative Process** | Involves constant iteration and improvement based on user feedback and changing needs | Can involve iterations based on usability testing and visual design improvements |
| **Outcome** | Successful UX results in satisfied users who find the product/service valuable and enjoyable | Successful UI results in an aesthetically pleasing and easy-to-use interface |
| **Examples** | Wireframes, user flows, prototypes, user personas | Visual design elements, style guides, color schemes, typography choices |

**Emotional Design Framework**

**Three Levels of Emotional Design**

**Visceral Design**

**Focus: Immediate, instinctive reactions to appearance**

* **Emphasis**: Aesthetic appeal, visual impact
* **Timeframe**: Immediate reaction upon first encounter
* **Importance:** Influences initial attraction and interest
* **Examples:** Product packaging, initial visual impression
* **Key Factors:** Appearance, aesthetics
* **Feedback:** Visual impact, initial impression
* **Goal:** Create immediate emotional response

**Behavioral Design**

**Focus: Usability and functionality during interaction**

* **Emphasis:** User experience, ease of use
* **Timeframe:** Experience during interaction with the product
* **Importance:** Determines satisfaction during usage
* **Examples**: User interface design, interaction flow
* **Key Factors:** Usability, functionality
* **Feedback:** Interaction experience, ease of use
* **Goal**: Provide seamless user experience

**Reflective Design**

**Focus: Long-term emotional connections and associations**

* **Emphasis:** Memories, emotional connections over time
* **Timeframe:** Emotional response over the product's lifespan
* **Importance:** Shapes long-term attachment and loyalty
* **Examples:** Brand loyalty, nostalgia, user testimonials
* **Key Factors:** Emotional connections, memories
* **Feedback:** Emotional associations, nostalgia
* **Goal: Foster** long-term emotional connection and loyalty

**The Seven Key Factors of User Experience**

**1. Useful**

**Definition:** If a product isn't useful to someone, why would you want to bring it to market?

**Key Points:**

* Products must have a clear purpose to compete in the market
* "Useful" is subjective and can include non-practical benefits like fun or aesthetic appeal
* Computer games or sculptures may be deemed useful even if they don't help accomplish traditional goals
* Usefulness is determined by the end user's perspective and needs

**2. Usable**

**Definition:** Usability is concerned with enabling users to effectively and efficiently achieve their end objective with a product.

**Key Points:**

* Products should be designed for human limitations (e.g., people typically have only 2 hands)
* Poor usability often characterizes first-generation products
* Example: Early MP3 players lost market share to the more usable iPod
* The iPod wasn't first to market but was the first truly usable MP3 player

**3. Findable**

**Definition:** The product must be easy to find, and for digital products, content within them must be easily discoverable.

**Key Points:**

* If users cannot find a product, they won't buy or use it
* Content organization is crucial (like newspaper sections: Sports, Entertainment, Business)
* Random organization creates frustrating user experiences
* Findability is vital across all product types

**4. Credible**

**Definition:** Users must be able to trust the product and its provider.

**Key Points:**

* Trust encompasses product functionality, durability, and information accuracy
* Modern users won't give second chances - competition is abundant
* Products must do what they promise and last for reasonable periods
* Information provided must be accurate and fit-for-purpose
* Lack of credibility drives users to competitors

**5. Desirable**

**Definition:** Beyond basic functionality, products should create emotional appeal and aspiration.

**Key Points:**

* Example: Both Skoda and Porsche make functional cars, but Porsche is more desirable
* Desirability is conveyed through branding, image, identity, aesthetics, and emotional design
* Desirable products encourage users to share and create desire in others
* Desirability often determines preference when multiple options meet basic needs

**6. Accessible**

**Definition:** Providing experiences accessible to users with full range of abilities, including disabilities.

**Key Points:**

* Includes users with hearing loss, impaired vision, motion impairments, or learning disabilities
* At least 19% of US population has a disability (likely higher in developing nations)
* Designing for accessibility often improves usability for everyone
* Legal obligation in many jurisdictions (EU and others)
* Failure to provide accessibility may result in fines

**7. Valuable**

**Definition:** Products must deliver value to both the business creating it and users buying/using it.

**Key Points:**

* Without value, initial success will eventually be undermined
* Value significantly influences purchasing decisions
* Example: $100 product solving $10,000 problem likely succeeds
* $10,000 product solving $100 problem much less likely to succeed
* Value proposition must be clear and compelling

**UX and UI in Video Game Design**

**User Interface (UI) in Games**

**1. Menus and Navigation**

* Include menus, buttons, and navigation systems
* Allow access to different game parts (settings, inventory, maps)
* Must be intuitive and responsive

**2. Heads-Up Display (HUD)**

* Displays essential gameplay information (health, ammunition, critical data)
* Should not obstruct player's view of game world
* Must be immediately readable and relevant

**3. Icons and Symbols**

* Visual elements representing actions, items, or status
* Make it easier for players to understand and interact
* Should be universally recognizable and consistent

**4. Consistency and Design**

* Consistent design maintains cohesive visual style
* Ensures players can understand and predict element functions
* Creates seamless experience across game sections

**5. Feedback and Responsiveness**

* UI elements must provide immediate feedback to player actions
* Players need awareness of action consequences
* Reduces confusion and improves engagement

**User Experience (UX) in Games**

**1. Onboarding and Tutorials**

* Effective processes to teach game mechanics, controls, objectives
* Should be engaging rather than overwhelming
* Progressive disclosure of complexity

**2. Player Engagement**

* Compelling storylines and interesting characters
* Well-balanced progression systems
* Contributes to positive overall experience

**3. Immersion and Atmosphere**

* Created through graphics, sound, and storytelling
* Enhances emotional connection to the game
* Maintains suspension of disbelief

**4. Flow and Pacing**

* Balancing difficulty levels appropriately
* Pacing progression to maintain interest
* Maintaining smooth gameplay flow

**5. Accessibility**

* Consideration for different player abilities and preferences
* Adaptability across different devices
* Ensures diverse audience can enjoy the game

**6. User Feedback and Iteration**

* Gathering feedback from playtesting sessions
* Incorporating feedback into design refinements
* Continuous improvement of user experience

**7. Multiplatform Considerations**

* Adapting design for different platforms (PC, consoles, mobile)
* Ensuring seamless experience across devices
* Platform-specific optimization while maintaining core experience

**10. Ergonomics in the Workplace**

**Definition and Importance**

Ergonomics is the science of designing workspaces and products to improve efficiency and reduce discomfort. It focuses on optimizing human well-being and overall system performance by designing environments that fit the user's needs.

**Key Ergonomic Concepts**

| **Term** | **Definition** |
| --- | --- |
| **Posture** | Way you hold your body while sitting, standing, or working |
| **Musculoskeletal Disorders (MSDs)** | Injuries affecting muscles, bones, and joints due to poor ergonomics |
| **Neutral Position** | Natural and balanced posture reducing strain during activities |
| **Repetitive Strain Injuries (RSI)** | Injuries from repeating same motions repeatedly |
| **Carpal Tunnel Syndrome** | Type of RSI affecting wrist and hand from repetitive movements |
| **Monitor Height** | Adjusting screen level to align with eyes for optimal viewing |
| **Keyboard Tray** | Sliding shelf holding keyboard at comfortable typing height |
| **Ergonomic Chair** | Chair designed to support posture during long sitting periods |
| **Eye Strain** | Discomfort from prolonged screen staring or improper lighting |
| **Footrest** | Platform supporting feet to maintain proper leg posture |
| **Workstation** | Area including desk, chair, computer, and tools |
| **Lighting** | Proper workspace illumination reducing eye strain |

**Common Workplace Health Issues**

**Computer-Related Problems**

* **Low Back Pain**: Very common in computer users (80% of users working 4+ hours complain of back pain)
* **Text Neck**: Condition from hunching over smartphones causing neck and shoulder strain
* **Eye Strain**: Result of prolonged screen use leading to blurry vision, fatigue, headaches
* **Repetitive Strain Injuries**: From repeated motions causing pain and discomfort
* **Poor Posture Effects**: Muscle stiffness, joint problems, spinal issues

**Risk Factors**

* Prolonged sitting
* Hunching over devices
* Carrying heavy bags
* Poor workstation setup
* Inadequate lighting
* Improper chair/desk height
* Lack of regular breaks

**Ergonomic Solutions**

**Workstation Setup**

* **Chair**: Adjustable height, proper back support, armrests
* **Desk**: Appropriate height, adequate space for legs
* **Monitor**: Eye level, arm's length distance, proper tilt
* **Keyboard/Mouse**: Neutral wrist position, ergonomic design
* **Lighting**: Adequate, reduced glare, adjustable

**Best Practices**

* **Take Regular Breaks**: Stand, stretch, move around frequently
* **Maintain Good Posture**: Straight spine, feet flat on floor, shoulders relaxed
* **Adjust Equipment**: Customize workstation to fit individual needs
* **Use Proper Techniques**: Correct lifting, carrying methods
* **Exercise Regularly**: Strengthen muscles, improve flexibility
* **Stay Hydrated**: Keep spinal discs healthy

**Treatment Approaches Study Results**

**Conventional vs Multidisciplinary Treatment**

**Study Findings** (44 participants, 22 per group):

* **Multidisciplinary approach** significantly better than conventional treatment
* **Pain intensity**: Significantly reduced in multidisciplinary group
* **Sick leave**: Only 19.19% in multidisciplinary group vs 63.63% in conventional group
* **Quality of life**: Significantly improved in multidisciplinary group

**Conventional Treatment**

* Orthopedic consultation
* Physical therapy

**Multidisciplinary Treatment**

* Orthopedic consultation
* Physiotherapy
* Ergonomics assessment
* Vitamin supplementation
* Diet planning
* Massage therapy
* Stress management

**French Vocabulary for Ergonomics**

* **Une étude interventionnelle prospective**: Prospective interventional study
* **La colonne vertébrale**: Spine
* **Les douleurs lombaires**: Low back pain
* **L'intensité de la douleur**: Pain intensity
* **Un congé maladie**: Sick leave
* **L'ergonomie**: Ergonomics
* **La kinésithérapie**: Physiotherapy
* **La supplémentation en vitamines**: Vitamin supplementation
* **La massothérapie**: Massage therapy
* **La gestion du stress**: Stress management

**11. Web Tracking and Privacy**

**Web Tracking Technologies**

**Cookies**

**Definition**: Small pieces of data stored on user's device to remember information about user's interactions with websites.

**Types**:

* **HTTP Cookies**: Store user preferences, session details, tracking data
* **Third-party Cookies**: Placed by domains other than current site, used for cross-site tracking

**Tracking Pixels (Web Beacons)**

**Definition**: Tiny, transparent images embedded in web pages or emails that send information back to server when loaded.

**Uses**:

* Track user interactions
* Measure email open rates
* Test email content effectiveness
* Retargeting advertisements

**Other Tracking Methods**

* **JavaScript Tracking**: Code snippets collecting user interaction data
* **Fingerprinting**: Gathering device/browser information to create unique identifier
* **Server Logs**: Recording IP addresses, user agents, visited URLs
* **Session/Local Storage**: HTML5 storage for tracking preferences and activities
* **Click Tracking**: Recording and analyzing user clicks and mouse movements

**Retargeting/Remarketing Process**

1. **User visits website** looking at products (e.g., shoes)
2. **Tracking pixel loads** with product information
3. **Data sent to ad network** (Google, Facebook, Amazon)
4. **Targeted ads created** based on user behavior
5. **User sees ads** for previously viewed products on other sites

**How Tracking Pixels Work**

* Website owner places tracking pixel on every page
* Pixel has URL that tells browser where to download image
* Extra data added to pixel URL (product views, shopping cart items)
* When page loads, pixel downloads with this data
* Ad network extracts and stores data for targeted campaigns

**Privacy Concerns and Solutions**

**User Perceptions**

* **60% of online buyers** notice retargeted ads
* **11% feel negatively** about seeing product-related ads
* **35% find retargeted ads creepy** (2018 study)
* Growing awareness of online privacy issues

**Privacy Protection Methods**

* **Browser Protections**: Safari, Firefox, Brave have built-in tracking protection
* **Browser Extensions**: Privacy Badger, Ghostery, ad blockers with privacy lists
* **Do Not Track (DNT)**: Browser setting requesting websites not to track
* **Opt-out Options**: User choice to decline tracking/advertisements

**Key Web Tracking Vocabulary**

| **Term** | **Definition** |
| --- | --- |
| **Cookies** | Small data pieces stored on device remembering user interactions |
| **IP Address** | Unique numerical label for devices on network |
| **Do Not Track (DNT)** | Browser request asking websites not to track user |
| **Analytics** | Collecting and analyzing user behavior data |
| **Tracking Pixels** | Code snippets tracking user interactions like email opens |
| **Behavioral Targeting** | Tailoring ads based on online behavior |
| **Privacy Policy** | Document explaining how user data is collected and used |
| **Session** | Short interaction between user and website |
| **User Agent** | Information about user's browser and device |
| **Geolocation** | Collecting data about user's physical location |
| **Opt-out** | Choice to decline participation in tracking/services |

**Web Tracking Phrasal Verbs**

* **Collect data**: Gather user information
* **Opt out**: Decline targeted advertisements
* **Track down**: Find source of problems
* **Zero in on**: Focus on specific demographics
* **Sign up**: Register for newsletters/offers
* **Pop up**: Appear (notifications asking permissions)
* **Scroll through**: Navigate browsing history
* **Clear out**: Remove unnecessary features
* **Log in**: Access accounts with credentials

**12. Mock Exam**

**Exam Structure**

| **Section** | **Points** | **Content** |
| --- | --- | --- |
| **Reading** | /10 | French comprehension of English technology text |
| **Theory + Grammar** | /40 | Dark patterns, CV tips, ergonomics, programming concepts |
| **Vocabulary** | /20 | Phrasal verbs, definitions, sentence construction |
| **Oral** | /30 | Speaking assessment |
| **Total** | /100 → /20 | Final grade conversion |

**Sample Questions from Mock Exam**

**Reading Comprehension (French Responses Required)**

Based on technology and spine health article:

1. Consequences of prolonged technology use on eye health
2. Why incorrect posture contributes to spinal health problems
3. Negative effects of excessive technology use on mental health
4. Effects on addiction
5. Three ergonomic tips for maintaining good posture

**Theory Questions**

1. **Dark Pattern Identification**: Analyze interface design for deceptive patterns
2. **CV Improvement**: Provide 4 tips for positive employer impression
3. **Ergonomics vs Aesthetics**: Discuss trade-offs in hardware/furniture design
4. **Python vs JavaScript**: Compare paradigms, features, uses
5. **E-government Apps**: Define and provide 3 examples
6. **Dark Pattern Scenarios**: Identify specific patterns in user stories

**Vocabulary Tasks**

1. **Phrasal Verb Replacement**: Replace underlined words with appropriate phrasal verbs
2. **Sentence Construction**: Create meaningful sentences demonstrating word understanding
3. **Concept Definition**: Define technical terms using own words

**Key Exam Topics Coverage**

* **Public Speaking**: Preparation, delivery, overcoming fear
* **Professional Communication**: CVs