



## PRELIMS

### Lesson 1: Introduction, HCI principles, and guidelines

#### **Introduction to Human-Computer Interaction (HCI)**

Human-Computer Interaction (HCI) is a field of study that focuses on the design, evaluation, and implementation of computer systems that are easy to use, effective, and enjoyable for humans. HCI principles and guidelines provide a framework for designing user-centered interfaces that meet the needs and expectations of users.

#### HCI Principles

Key principles of HCI include:

1. User-centered design: The design of a system should be driven by the needs, wants, and abilities of the users.
2. Task-centered design: The system should support the users in performing their tasks effectively and efficiently.
3. Usability: The system should be easy to learn, use, and remember.
4. Accessibility: The system should be accessible to users with a wide range of abilities and disabilities.
5. Feedback: The system should provide users with feedback about their actions and the state of the system.
6. Consistency: The system should use consistent design elements and terminology.
7. Error prevention: The system should be designed to prevent errors from occurring, and to help users recover from errors when they do occur.
8. Efficiency: The system should allow users to perform tasks quickly and efficiently.
9. Satisfaction: The system should be enjoyable to use and satisfy the users' needs.



## HCI Guidelines

HCI guidelines are specific recommendations for designing user-centered interfaces. They are based on research and empirical evidence, and they provide practical guidance for designers and developers. Some examples of HCI guidelines include:

- Nielsen's Heuristics: A set of 10 principles for designing user-friendly interfaces.
- Norman's Principles of Interaction: A set of seven principles for designing natural and intuitive interfaces.
- Shneiderman's Eight Golden Rules of Interaction Design: A set of eight guidelines for designing effective and usable interfaces.
- GOMS: A model for understanding how users interact with computer systems, and for designing interfaces that support efficient user interaction.

These principles and guidelines can help to ensure that computer systems are designed to be easy to use, effective, and enjoyable for humans.

### 1. Nielsen's Heuristics:

Example: A progress bar that shows how long a file is taking to download (Visibility of system status).

Simple explanation: These are ten usability principles that guide intuitive and user-friendly interface design. They emphasize things like keeping users informed, using familiar language, allowing easy undo/redo, and minimizing memory load.

### 2. Norman's Principles of Interaction:

Example: A door handle shaped like a lever you push or pull (Natural mapping).

Simple explanation: These principles focus on making interfaces predictable and understandable. They advocate for using natural mappings between controls and actions, providing feedback, and offering undo/redo options.

### 3. Schneiderman's Eight Golden Rules of Interaction Design:



Example: A website search bar with auto-suggestions as you type (Strive for consistency). Simple explanation: These rules emphasize efficiency and ease of use. They suggest things like minimizing user effort, making actions reversible, and providing good visual feedback.

4. GOMS: Example: Selecting a word in a document by clicking and dragging (Goal, Operator, Method, Selection). Simple explanation: GOMS is a model for predicting human-computer interaction. It breaks down actions into goals, operators (mental actions), and methods (physical actions). By analyzing GOMS, designers can optimize interfaces for faster and more efficient use. In essence, these principles offer different perspectives on designing user interfaces for optimal usability and ease of use. Choosing and applying them strategically can improve your designs and create enjoyable user experiences. Remember, these are just simplified examples. Each principle has its nuances and complexities that you can explore further for deeper understanding.

## Lesson 2: Design Process

The **design process** is a systematic approach to creating solutions that are both functional and aesthetically pleasing. It involves a series of steps that are typically iterative, meaning that they are repeated and refined as the design develops.

The design process can be divided into the following stages:

1. **Define the problem:** The first step is to clearly define the problem that the design needs to solve. This involves understanding the user's needs, the context of use, and the limitations of the project.
2. **Gather information:** Once the problem has been defined, the next step is to gather information about the target audience, the competition, and the available resources. This information can be gathered through user research, competitive analysis, and market research.
3. **Ideate:** The ideation stage is where the creative process begins. Designers will brainstorm and generate as many ideas as possible, without worrying about feasibility or cost. This stage is often referred to as "thinking outside the box."
4. **Sketch and refine:** Once the initial ideas have been generated, they need to be sketched and refined. This can be done on paper, using digital tools, or a combination of both. The goal of this stage is to develop a concrete representation of the design ideas.



5. **Create prototypes:** Prototypes are low-fidelity representations of the design that can be used to test and gather feedback. There are many different types of prototypes, including paper prototypes, wireframes, and interactive prototypes.
6. **Test and iterate:** The design process is iterative, meaning that it is repeated and refined as the design develops. This is where testing and feedback play a crucial role. Designers will test the prototypes with users to gather feedback on usability, functionality, and aesthetics. This feedback will then be used to iterate on the design.
7. **Develop and deploy:** Once the design has been finalized, it is ready to be developed and deployed. This may involve coding, writing, and other technical tasks.
8. **Evaluate and maintain:** Once the design has been deployed, it is important to evaluate its performance and make sure that it is meeting the needs of the users. This may involve gathering usage data and conducting user interviews. It is also important to maintain the design over time, as user needs and technology evolve.

## Lesson 3: Interaction design basics

**Interaction design** (IxD) is the practice of designing the interactions between people and products or services. It encompasses a variety of disciplines, including psychology, sociology, computer science, and graphic design. The goal of interaction design is to create products and services that are easy to use, effective, and enjoyable.

Here are some of the **key principles** of interaction design:

1. **User-centered design:** The design of a product or service should be driven by the needs, wants, and abilities of the users. This means understanding who the users are, what they are trying to do, and how they will use the product or service.
2. **Task-centered design:** The product or service should support the users in performing their tasks effectively and efficiently. This means understanding the tasks that users need to perform, and designing the product or service to make it easy to complete those tasks.
3. **Usability:** The product or service should be easy to learn, use, and remember. This means designing interfaces that are clear, consistent, and intuitive.



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

**4. Accessibility:** The product or service should be accessible to users with a wide range of abilities and disabilities. This means considering the needs of users with physical, cognitive, and sensory impairments.

**5. Feedback:** The product or service should provide users with feedback about their actions and the state of the system. This means providing clear and timely feedback that helps users understand what is happening and what they need to do next.

**6. Consistency:** The product or service should use consistent design elements and terminology. This means using a consistent style guide and making sure that the product or service is easy to learn and use.

**7. Error prevention:** The product or service should be designed to prevent errors from occurring, and to help users recover from errors when they do occur. This means using techniques such as input validation, progressive disclosure, and error messages.

**8. Efficiency:** The product or service should allow users to perform tasks quickly and efficiently. This means minimizing the amount of time and effort required to complete tasks.

**9. Satisfaction:** The product or service should be enjoyable to use and satisfy the users' needs. This means creating products and services that are aesthetically pleasing, engaging, and motivating.

Here are some of the **key tools and techniques** used in interaction design:

**1. User research:** User research is the process of understanding the needs, wants, and behaviors of users. This can be done through a variety of methods, such as interviews, surveys, and usability testing.

**2. Information architecture:** Information architecture is the process of organizing and structuring information in a way that is easy for users to find and understand. This includes creating sitemaps, taxonomies, and user flows.

**3. User interface (UI) design:** UI design is the process of designing the visual appearance and interaction of a product or service. This includes creating wireframes, prototypes, and mockups.



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

**4. User experience (UX) design:** UX design is the process of designing the overall experience of using a product or service. This includes considering the user's needs, wants, and expectations at every stage of the design process.

**5. Prototyping:** Prototyping is the process of creating low-fidelity or high-fidelity models of a product or service. This allows users to test and interact with the product or service before it is finalized.

**6. Usability testing:** Usability testing is the process of observing users as they interact with a product or service. This can identify usability problems and help to improve the design.

Interaction design is a complex and challenging field, but it is also a very rewarding one. By following the principles and using the tools and techniques described above, interaction designers can create products and services that are easy to use, effective, and enjoyable for everyone.

<https://bootcamp.uxdesign.cc/10-things-you-have-to-learn-if-you-want-to-be-a-ui-ux-designer-6919c939fae6>



## Lesson 4: HCI in the software process

**Human-Computer Interaction (HCI)** plays a crucial role throughout the software development process, ensuring that the resulting software is user-centered, usable, and enjoyable. Here's an overview of how HCI is integrated into the various phases of software development:

1. **Requirements Analysis:** During requirements gathering, HCI practitioners employ techniques like user interviews, focus groups, and surveys to understand user needs, goals, and pain points. This information is then used to define user-centered requirements that align with the software's purpose and target audience.
2. **Design:** In the design phase, HCI experts collaborate with software designers to create user interfaces (UIs) that are intuitive, consistent, and visually appealing. They consider factors like user interaction patterns, accessibility needs, and cultural sensitivities to ensure the UI is easy to learn and use.
3. **Development:** During development, HCI practitioners work with software developers to implement the UI design and ensure it adheres to HCI principles. They may also conduct usability testing to identify and rectify any usability issues before the software is released.
4. **Testing:** Usability testing is an integral part of the software development process, and HCI experts play a key role in planning, conducting, and analyzing usability tests. They observe users interacting with the software, identify usability problems, and provide feedback to developers for improvement.
5. **Deployment and Maintenance:** After the software is deployed, HCI continues to be relevant. HCI practitioners may track user feedback, conduct post-release usability testing, and analyze usage data to identify areas for further improvement. They also contribute to ongoing maintenance and updates to ensure the software remains user-friendly and up-to-date.

In summary, HCI is not just a separate discipline but rather an integral part of the entire software development process. By incorporating HCI principles and practices throughout the development lifecycle, software companies can create products that are not only functional but also enjoyable and effective for their users.



## Lesson 5: Design rules

**Design rules** are a set of guidelines that help designers create consistent, effective, and aesthetically pleasing designs. They are based on principles of visual hierarchy, balance, proportion, contrast, and unity. Here are some of the most important design rules:

Design Diaries @TwoXStudio

# GRAPHIC DESIGN PRINCIPLES

The infographic is titled "GRAPHIC DESIGN PRINCIPLES" in large, bold, black letters at the top center. It features a grid of nine squares, each containing a graphic element and a corresponding principle name below it. The principles are: BALANCE (two vertical yellow rectangles), UNITY (a 3x3 grid of yellow rectangles), CONTRAST (a square divided into four quadrants with yellow and grey colors), RHYTHM (vertical yellow stripes of varying widths), HIERARCHY (horizontal grey stripes of varying widths), ALIGNMENT (a vertical line dividing a yellow rectangle in half), PROPORTION (a large yellow square next to a much smaller one), VARIETY (a square containing a yellow square, a grey circle, a grey triangle, and another yellow square), and EMPHASIS (a 2x2 grid where the bottom-right square is grey while others are yellow).

PRINCIPLE	DESCRIPTION
BALANCE	Two vertical yellow rectangles.
UNITY	A 3x3 grid of yellow rectangles.
CONTRAST	A square divided into four quadrants with yellow and grey colors.
RHYTHM	Vertical yellow stripes of varying widths.
HIERARCHY	Horizontal grey stripes of varying widths.
ALIGNMENT	A vertical line dividing a yellow rectangle in half.
PROPORTION	A large yellow square next to a much smaller one.
VARIETY	A square containing a yellow square, a grey circle, a grey triangle, and another yellow square.
EMPHASIS	A 2x2 grid where the bottom-right square is grey while others are yellow.

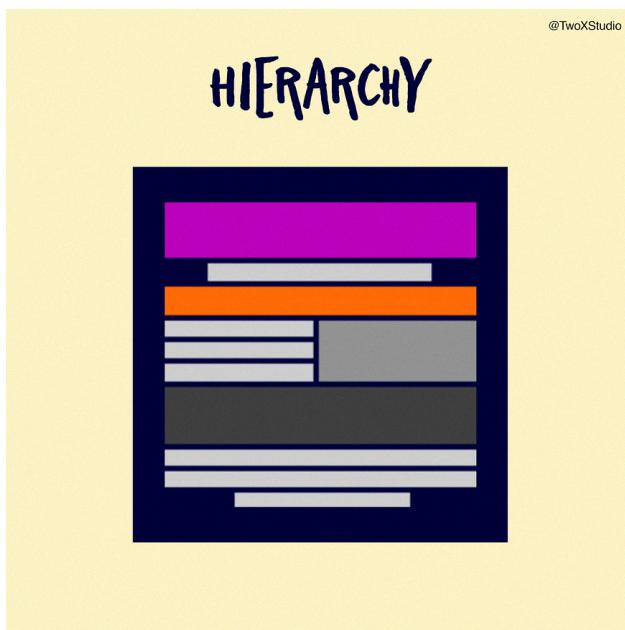


Telefax No.: (074) 442-3071

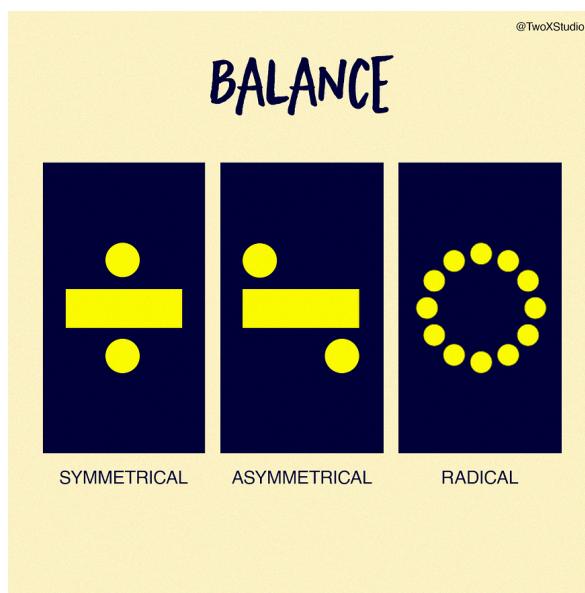
Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

1. **Hierarchy:** Use size, color, contrast, and placement to establish a hierarchy of information, guiding the viewer's eye to the most important elements first.



2. **Balance:** Create a visually balanced composition by distributing elements evenly or using asymmetrical balance to create visual interest.

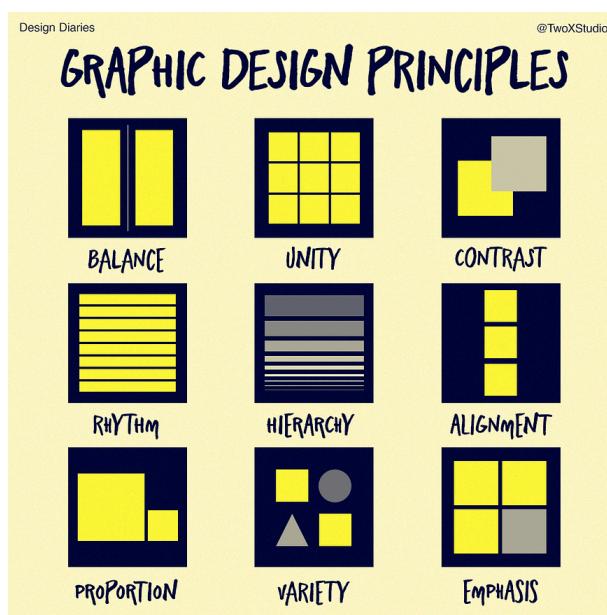




3. **Proportion:** Use consistent proportions between elements to establish harmony and visual coherence.

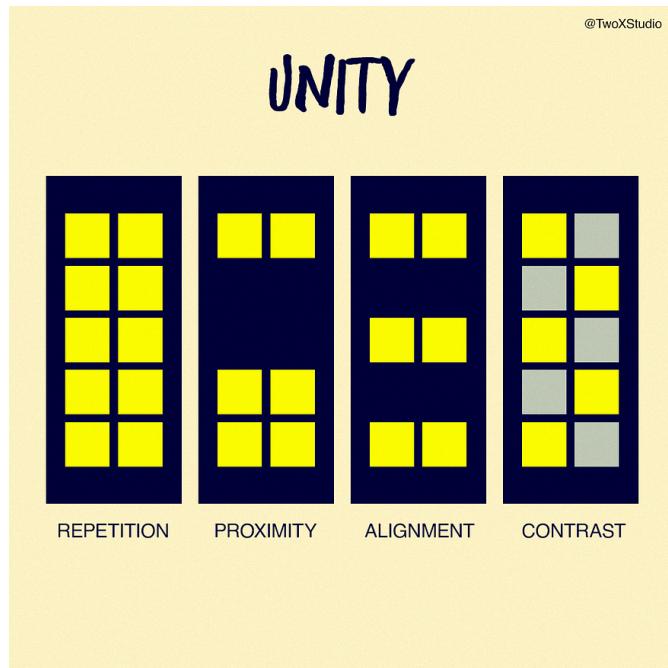


4. **Contrast:** Employ contrast in color, value, and texture to make elements stand out and differentiate them from each other.





5. **Unity:** Create a unified design by using consistent visual elements, such as color schemes, typography, and design motifs.





**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

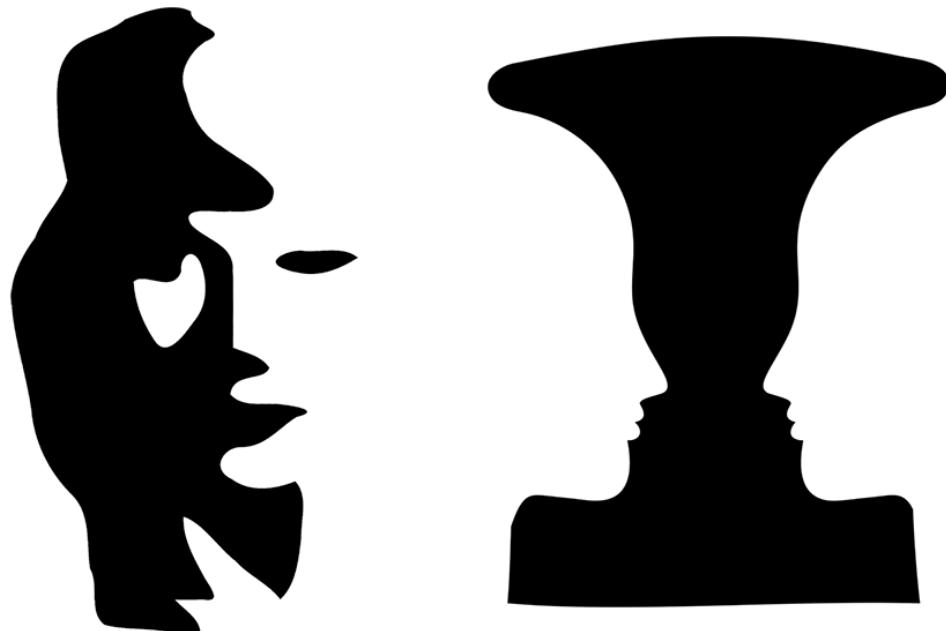
Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

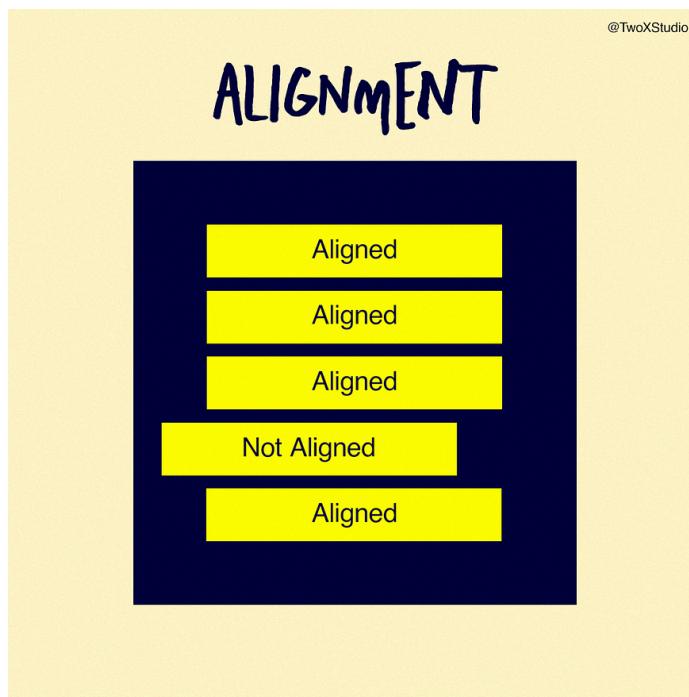
Additional design rules include:

1. **Negative space (white space):** Use negative space to create breathing room around elements, making them more prominent and improving readability.





2. **Alignment:** Align elements consistently to create order and organization within the design.



3. **Repetition:** Repeat visual elements to create a sense of coherence and rhythm.





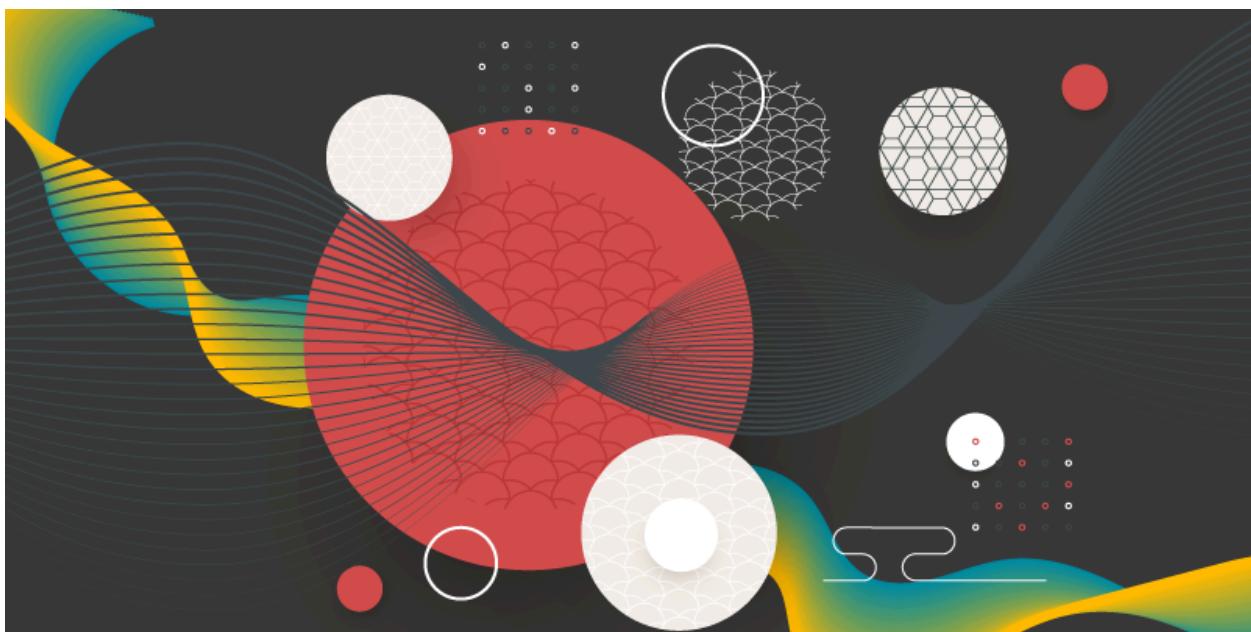
UNIVERSITY OF  
**Baguio**  
SCHOOL OF INFORMATION TECHNOLOGY  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

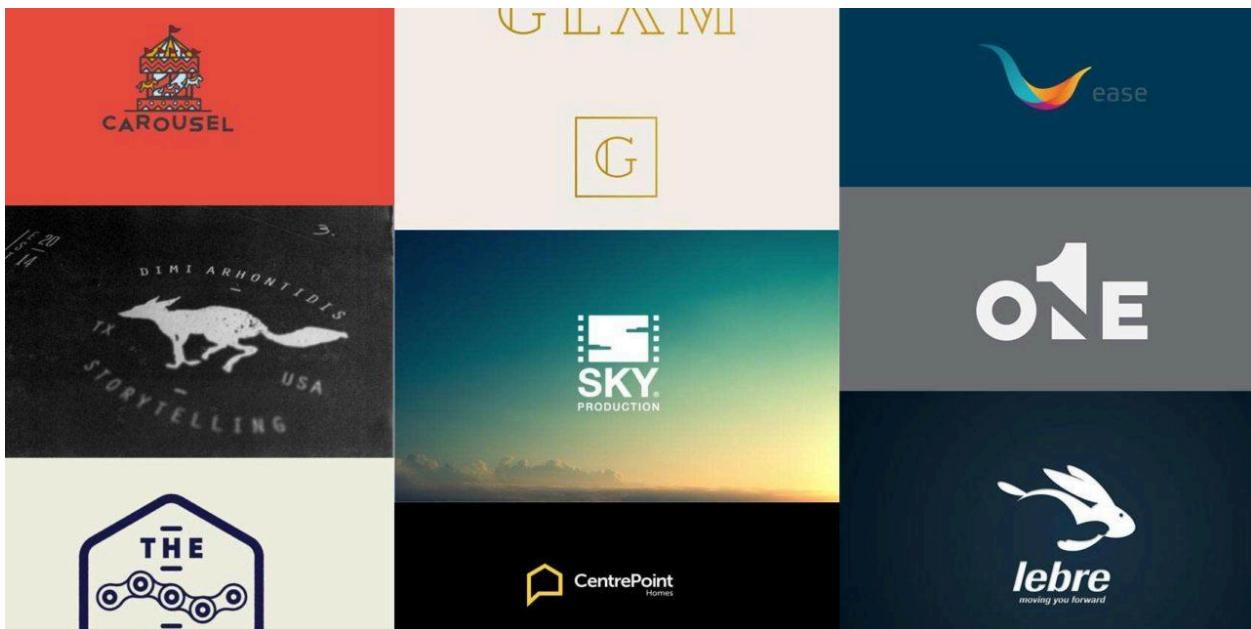
Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

4. **Movement:** Guide the viewer's eye through the design using visual cues like lines, shapes, and color gradients.

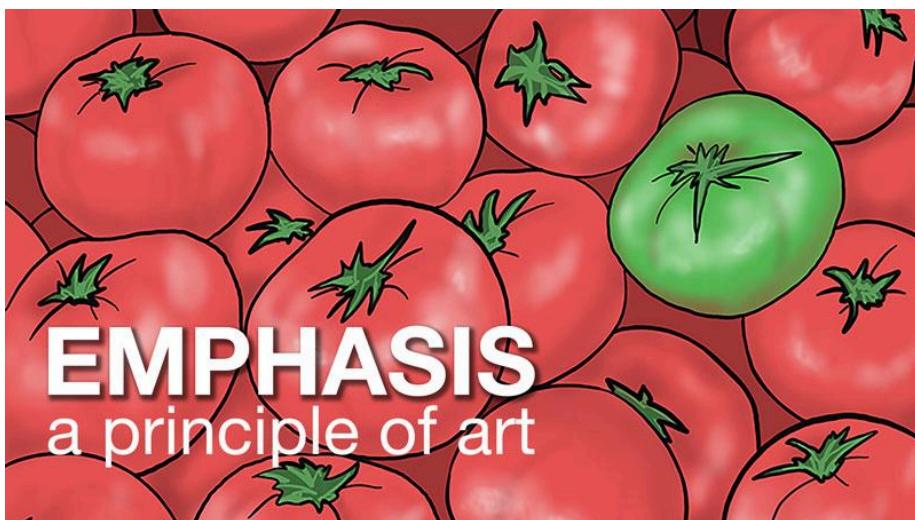


5. **Simplicity:** Strive for simplicity and avoid clutter to enhance clarity and focus.

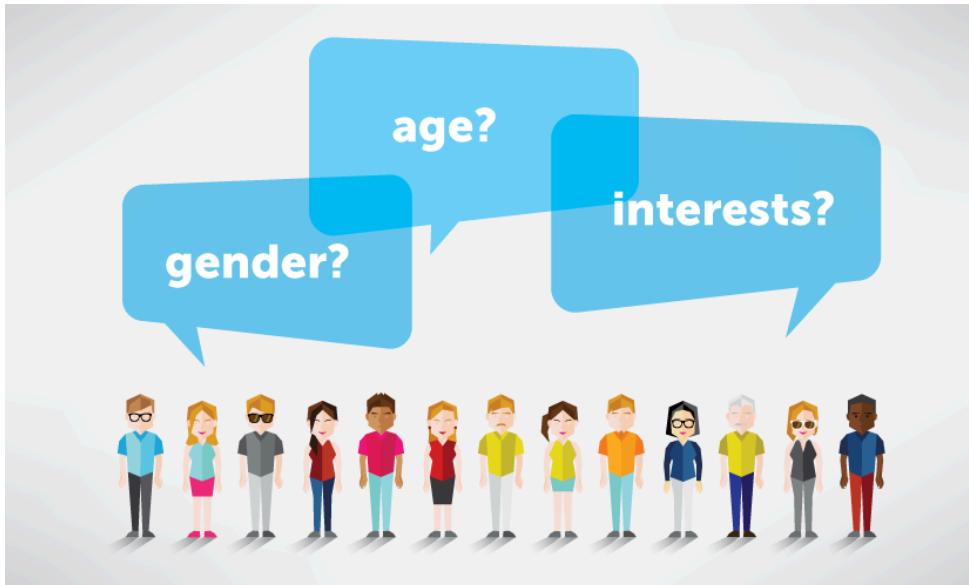




6. **Emphasize the purpose:** Clearly communicate the purpose of the design and ensure it aligns with the intended message.



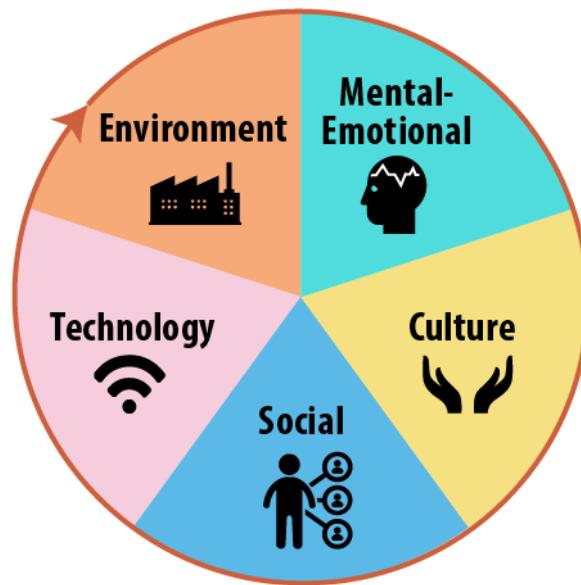
7. **Consider the target audience:** Tailor the design to the preferences and needs of the target audience.



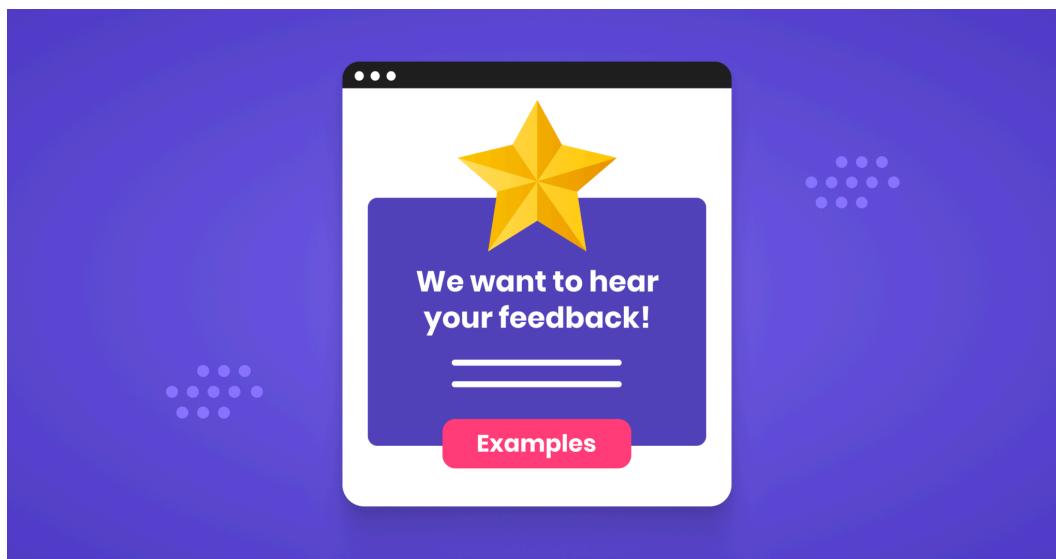


8. **Context is crucial:** Design with the context in mind, considering how the design will be used and viewed.

### 360° OF CONTEXT

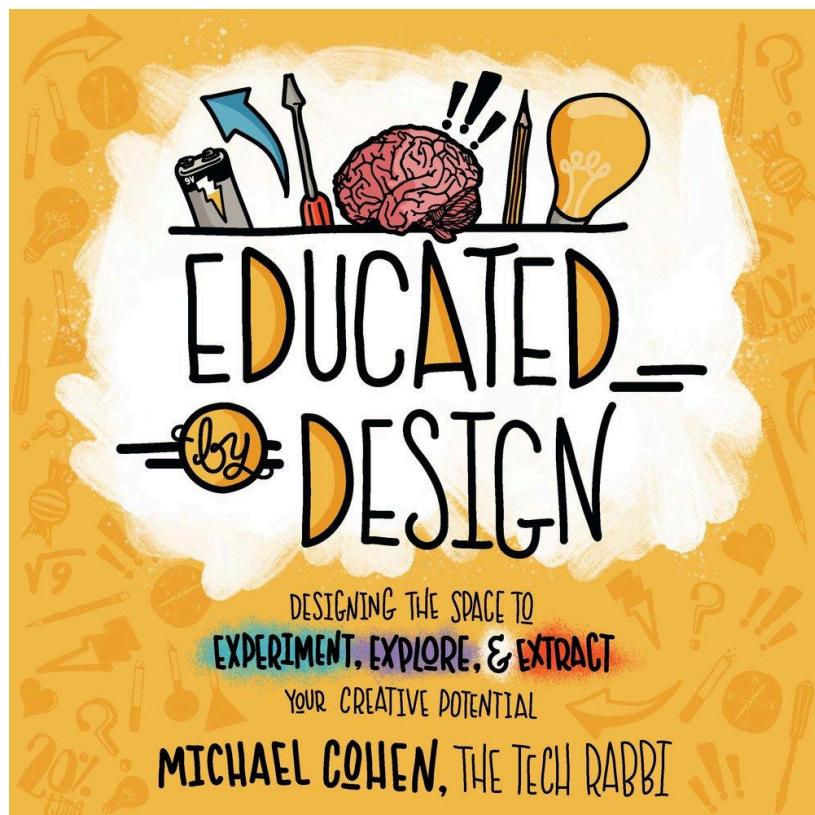


9. **Seek feedback:** Gather feedback from others to identify areas for improvement and refine the design.





10. **Experiment and explore:** Embrace experimentation and exploration to discover new design possibilities and enhance creativity.



By following these design rules and principles, designers can create visually appealing, effective, and user-centered designs that resonate with their target audience.



## Lesson 6: Implementation support

**Implementation support** is a planned approach to **integrate new or upgraded software** or systems into the existing workflow of an organizational structure to help ensure the success of a business' overall system. It encompasses a wide range of activities aimed at assisting organizations in effectively **implementing and adopting** new technologies, processes, or practices.

# IMPLEMENT

How to develop and launch an optimised user experience:

1

Produce guidelines for implementation.



2

Develop and launch final product or service.



3

Deploy a measurement framework.



4

Continue to learn and iterate.



5

Update in line with expert guidelines.

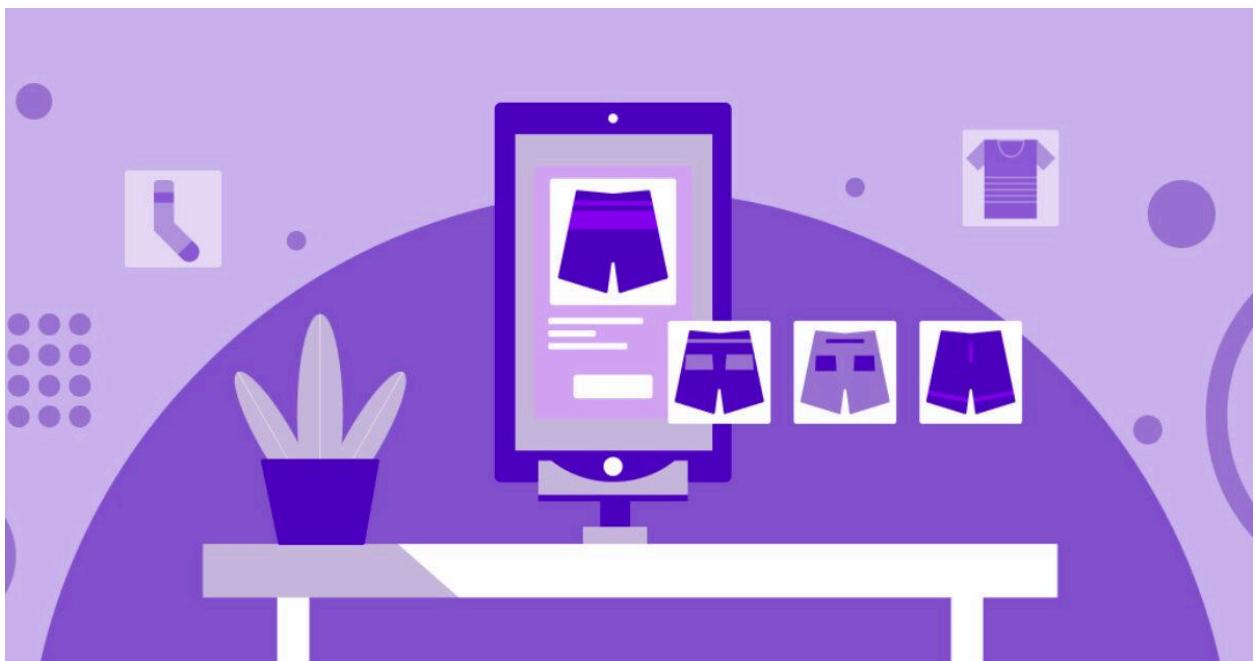


 System Concepts

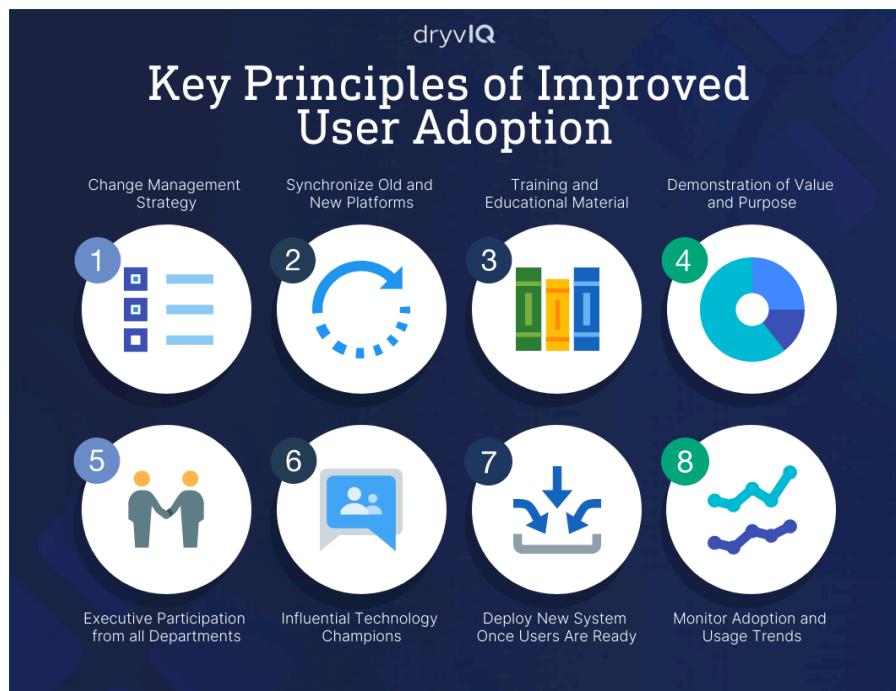


Key Objectives of Implementation Support:

1. **Smooth Integration:** Ensure seamless integration of new technologies or systems into existing infrastructure and processes, minimizing disruption to ongoing operations.



2. **User Adoption:** Facilitate user adoption and acceptance of new technologies or systems, providing training, support, and guidance to help users adapt and become proficient.



3. **Change Management:** Manage organizational change effectively, addressing resistance, addressing concerns, and fostering a culture of acceptance for the new implementation.





4. **Performance Optimization:** Optimize the performance of the newly implemented technologies or systems, ensuring they meet the organization's needs and deliver expected benefits.



5. **Continuous Improvement:** Implement continuous improvement cycles, gathering feedback, identifying areas for improvement, and refining the implementation to maximize effectiveness.





### Types of Implementation Support Services:

1. **Needs Assessment:** Conduct a comprehensive assessment to understand the organization's needs, goals, and challenges related to the implementation.
2. **Planning and Design:** Develop a detailed implementation plan, outlining the steps, resources, and timelines required for the successful implementation.
3. **Training and Education:** Provide training and educational materials to equip users with the necessary skills and knowledge to operate the new technologies or systems effectively.
4. **Change Management:** Develop and implement a change management strategy to address organizational resistance, communicate effectively, and gain user buy-in.
5. **Data Migration and Integration:** Migrate data from existing systems to the new ones, ensuring data integrity and compatibility.
6. **Testing and Validation:** Conduct rigorous testing and validation to identify and resolve any bugs, glitches, or compatibility issues before full-scale deployment.
7. **Post-Implementation Support:** Provide ongoing support after the initial implementation to address user queries, troubleshoot problems, and make adjustments as needed.
8. **Evaluation and Measurement:** Evaluate the success of the implementation, measure its impact on organizational performance, and make recommendations for further improvement.

### Benefits of Effective Implementation Support:

1. **Reduced Costs and Risks:** Minimize the risk of implementation failure, saving time, money, and resources.
2. **Improved User Adoption:** Enhance user satisfaction, productivity, and acceptance of new technologies or systems.
3. **Increased Efficiency:** Streamline operations, improve workflow efficiency, and reduce errors.
4. **Enhanced Competitive Advantage:** Gain a competitive edge by leveraging new technologies and processes effectively.
5. **Organizational Agility:** Foster a culture of adaptability and continuous improvement, enabling the organization to embrace change and respond to evolving needs.



**UNIVERSITY OF  
Baguio**

**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

In conclusion, implementation support plays a crucial role in ensuring the successful adoption and utilization of new technologies, processes, or practices within organizations. By providing comprehensive support throughout the implementation lifecycle, organizations can maximize the benefits of their investments and achieve their desired outcomes.

<https://www.m-inc.com/services/implementation-support-services/#:~:text=WHAT%20IS%20IMPLEMENTATION%20SUPPORT%3F,of%20a%20business%20overall%20system>.



## Lesson 7: Universal Design

**Universal Design (UD)** is a design philosophy that aims to create products, environments, and services that are usable by everyone. It is based on the principle of designing for the widest possible range of users, including people of all ages, abilities, and backgrounds.

Key Principles of Universal Design:

1. **Equitable Use:** The design is usable and accessible to people with diverse abilities.
2. **Flexibility in Use:** The design accommodates a wide range of individual preferences and abilities.
3. **Simple and Intuitive Use:** The design is easy to understand, operate, and learn to use.
4. **Perceptible Information:** The design provides clear and unambiguous information about its operation.
5. **Tolerance for Error:** The design minimizes hazards and the negative consequences of error.
6. **Low Physical Effort:** The design can be used efficiently and comfortably with a minimum of physical effort.
7. **Size and Space for Approach and Use:** The design can be approached, reached, operated, and used comfortably by people with a wide range of body sizes and postures.
8. **Adaptive to Individual Needs and Abilities:** The design allows personalization to accommodate individual preferences and abilities.
9. **Tolerance for Limited Skill and Knowledge:** The design can be used effectively by people with limited skills or knowledge.

Applications of Universal Design:

1. **Products:** Products such as computers, appliances, and toys can be designed for universal use.
2. **Environments:** Buildings, public spaces, and transportation systems can be designed for universal accessibility.



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

3. **Services:** Services such as education, healthcare, and employment can be designed for universal usability.

#### Benefits of Universal Design:

1. **Greater Accessibility:** Universal design makes products, environments, and services accessible to a wider range of people.
2. **Reduced Costs:** Universal design can reduce the need for specialized products and services.
3. **Improved Usability:** Universal design can improve the usability of products and services for everyone.
4. **Greater Equality:** Universal design promotes equality and inclusion for all people.

#### Challenges of Universal Design:

1. **Balancing Cost and Function:** Universal design may require additional investment in design and development.
2. **Changing User Needs:** User needs and preferences can change over time, requiring ongoing design updates.
3. **Lack of Awareness:** There is still a lack of awareness of universal design among designers, manufacturers, and consumers.

#### Conclusion

**Universal design** is a valuable approach to creating products, environments, and services that are usable by everyone. By following the principles of universal design, we can create a more inclusive and equitable world for all.

#### Sources

1. [books.google.com/books?id=\\_N8xvuf9rcC](https://books.google.com/books?id=_N8xvuf9rcC)



2. [es.scribd.com/document/294725189/Accessibility-of-Housing-A-Handbook-of-Inclusive-Affordable-Housing-Solutions-for-Persons-with-Disabilities-and-Older-Persons](http://es.scribd.com/document/294725189/Accessibility-of-Housing-A-Handbook-of-Inclusive-Affordable-Housing-Solutions-for-Persons-with-Disabilities-and-Older-Persons)

## Lesson 8: Conceptualizing Interaction:

### Models, Interface Metaphors, Interaction types

**Conceptualizing interaction** is the process of understanding and defining how users will interact with a product or service. This involves identifying the user's goals, the tasks they need to accomplish, and the tools or technologies they will use. It also involves considering the context in which the interaction will take place, such as the physical environment and the user's cultural background.

#### Models

**Models** are used to represent and understand interactions in a variety of ways. Some common models include:

- **Mental models:** These are the mental representations that users have of how things work. They can be based on experience, analogy, or instruction.
- **Conceptual models:** These are abstract representations of the interaction that are designed to be easy to understand. They can be used to communicate the design to stakeholders and to develop prototypes.
- **Task models:** These are detailed representations of the tasks that users need to accomplish. They can be used to identify potential usability problems and to design solutions.



- **User flow models:** These are diagrams that show the steps that users take to complete tasks. They can be used to identify potential bottlenecks and to improve the efficiency of the interaction.

## Interface Metaphors

**Interface metaphors** are a way of using familiar concepts to make a new concept easier to understand. For example, a computer desktop is based on the metaphor of a real-world desk. This metaphor makes it easier for users to understand how to organize and manipulate files.

Interface metaphors can be very effective, but they can also be misleading. For example, a trash can metaphor might lead users to believe that deleted files can be easily recovered. It is important to choose metaphors carefully and to make sure that they are consistent with the underlying functionality of the interaction.

## Interaction Types

There are many different types of interaction, but some of the most common include:

- **Command-driven interaction:** This type of interaction is based on the user issuing commands to the system. This can be done through a variety of means, such as typing, clicking on buttons, or using voice commands.
- **Data-driven interaction:** This type of interaction is based on the user providing data to the system. This data can be used to generate reports, perform calculations, or control devices.
- **Direct manipulation interaction:** This type of interaction is based on the user directly manipulating objects on the screen. This can be done using a mouse, trackpad, or touchscreen.
- **Conversational interaction:** This type of interaction is based on the user having a natural conversation with the system. This can be done using voice commands or text chat.

## Conclusion



Telefax No.: (074) 442-3071

Website: [www.ubaguo.edu](http://www.ubaguo.edu)

E-mail Address: [sit@e.ubaguo.edu](mailto:sit@e.ubaguo.edu)

Conceptualizing interaction is a critical step in the design process. By understanding the user's goals, tasks, and context, and by using models, metaphors, and interaction types, designers can create interactions that are easy to use, effective, and enjoyable.

**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguo.edu](http://www.ubaguo.edu)

E-mail Address: [sit@e.ubaguo.edu](mailto:sit@e.ubaguo.edu)

## MIDTERMS

### Lesson 9. Cognition and Perception

**Cognition and perception** are two closely related processes that play a vital role in how we interact with the world around us. Cognition refers to the mental activities involved in understanding, learning, and remembering information. Perception, on the other hand, is the process of taking in information from our senses and interpreting it.

**Cognition and perception** are interdependent, and they work together to help us make sense of the world. Our senses provide us with raw data, but it is our cognition that



allows us to organize, interpret, and understand this information. For example, when we see an object, our visual system takes in light reflected from the object and sends this information to our brains. Our brains then process this information, and we recognize the object as a chair, table, or something else.

**Our cognition and perception** are also influenced by our past experiences and knowledge. This is why someone who is familiar with cars will be able to identify a car at a glance, while someone who is not familiar with cars may not be able to tell the difference between a car and a truck.

**Cognition and perception** are also affected by our emotions and motivations. For example, if we are excited about something, we may be more likely to notice and remember positive information. Conversely, if we are anxious, we may be more likely to notice and remember negative information.

The relationship between cognition and perception is complex and constantly evolving. As we learn more about the brain and the mind, we are better able to understand how these two processes work together to allow us to experience and interact with the world around us.

Here are some examples of how cognition and perception interact:

1. **Selective attention:** We can only focus on a limited amount of information at a time. Our brains use a process called selective attention to filter out irrelevant information and focus on what is most important.
  
  
  
2. **Schemas:** Schemas are mental representations of how the world works. They help us to organize and interpret new information.
  
  
  
3. **Heuristics:** Heuristics are mental shortcuts that we use to make decisions quickly and efficiently. They can be helpful, but they can also lead to errors.



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

**4. Confirmation bias:** Confirmation bias is the tendency to seek out information that confirms our existing beliefs and to ignore information that contradicts them.

**5. Illusions:** Illusions are visual or auditory experiences that are caused by our brains misinterpreting information from our senses. They can be caused by a variety of factors, such as physical limitations, expectations, and past experiences.

**Cognition and perception** are essential for our ability to interact with the world around us. By understanding how these two processes work together, we can develop better products, services, and experiences that are more user-friendly and enjoyable.

Lesson 10. Social Interaction:



- a. Introduction
- b. Types of Social Interaction
- c. Social interaction in UI design

## Social Interaction: A Comprehensive Overview

### Introduction

**Social interaction**, the cornerstone of human connection, is the exchange of information, ideas, and emotions between individuals or groups. It's the foundation of our social fabric, enabling us to form relationships, collaborate, and build communities. Social interaction is not limited to face-to-face encounters; it encompasses a wide spectrum of communication modes, including verbal, nonverbal, and mediated interactions.

### Types of Social Interaction

Social interaction can be categorized into various types based on its nature, purpose, and context:

1. **Dyadic Interaction:** This involves two individuals engaging in a direct, one-on-one exchange.
2. **Small Group Interaction:** This involves a small number of individuals, typically 3-5, engaged in a focused discussion or activity.
3. **Large Group Interaction:** This involves a larger number of individuals, ranging from a few dozen to thousands, engaged in a wider-scale event or discussion.
4. **Formal Interaction:** This occurs within structured settings, following established rules and protocols, such as business meetings or academic lectures.
5. **Informal Interaction:** This occurs in more casual settings, characterized by spontaneity and a relaxed atmosphere, such as conversations among friends or family.
6. **Verbal Interaction:** This involves the exchange of information through spoken or written language.



7. **Nonverbal Interaction:** This involves the exchange of information through nonverbal cues, such as facial expressions, body language, and tone of voice.
8. **Mediated Interaction:** This involves the use of technology to facilitate communication, such as through video conferencing or social media platforms.

## **Social Interaction in UI Design**

The principles of social interaction play a crucial role in user interface (UI) design, influencing how users interact with digital products and services. UI designers strive to create interfaces that foster meaningful social connections, facilitate collaboration, and enhance the overall user experience.

1. **Social Cues and Feedback:** UI elements can incorporate social cues, such as avatars, emoticons, and status indicators, to provide feedback and enhance the sense of social presence.
2. **Conversational Interfaces:** Chatbots and other conversational interfaces can mimic natural language interactions, enabling users to engage with products or services in a more natural and intuitive way.
3. **Collaborative Tools:** Features like shared workspaces, real-time editing, and group communication tools can facilitate collaboration among users, enabling them to work together effectively.
4. **Social Networking Integration:** Integration with social media platforms allows users to connect with their existing social circles within the digital product, enhancing engagement and community building.
5. **Accessibility and Inclusiveness:** UI design should consider the needs of diverse users, including those with disabilities, to ensure that social interactions are accessible and inclusive for all.

By incorporating these considerations, UI designers can create interfaces that not only support functionality but also foster meaningful social connections and enhance the overall user experience.



## Lesson 11. Designing Experience

**Designing experiences** is an intricate and multifaceted process that encompasses crafting meaningful interactions between users and products, services, or environments. It involves a deep understanding of human behavior, empathy for user needs, and the ability to translate abstract concepts into tangible experiences.

Key Principles of Experience Design:

1. **User-Centered Design:** Experience design should be driven by user needs, wants, and motivations. It's about understanding the context in which users will interact with the experience and tailoring it to their specific requirements.
2. **Holistic Approach:** Experience design considers all aspects of the user's journey, from initial contact to ongoing engagement. It encompasses not just the physical product or service but also the surrounding environment, interactions with other users, and the overall emotional impact.
3. **Storytelling:** Experience designers are storytellers who weave narratives that engage users and evoke emotions. They create a sense of journey, anticipation, and resolution, making the interaction memorable and meaningful.
4. **Empathy:** Experience designers must put themselves in the shoes of their users, empathizing with their needs, desires, and frustrations. This empathy is essential for creating experiences that resonate with users and address their pain points.
5. **Iteration and Prototyping:** Experience design is an iterative process, involving continuous refinement and improvement. Prototypes, ranging from low-fidelity sketches to high-fidelity simulations, are used to test and validate design decisions, ensuring the experience meets user expectations.



### Stages of the Experience Design Process:

1. **Research and Planning:** This initial stage involves understanding the user's needs, context, and motivations. It includes conducting user research, creating personas, and defining user goals and objectives.
2. **Conceptualization and Ideation:** This stage involves brainstorming and generating ideas for the experience. It includes exploring different concepts, sketching ideas, and developing scenarios to visualize the user journey.
3. **Prototyping and Testing:** This stage involves creating prototypes of the experience to test and refine design decisions. It includes feedback loops from users, stakeholders, and experts to ensure the experience is effective and enjoyable.
4. **Implementation and Delivery:** This stage involves bringing the experience to life. It includes developing the product, service, or environment, ensuring it aligns with the design vision and meets user expectations.
5. **Evaluation and Measurement:** This ongoing stage involves evaluating the impact of the experience and identifying areas for improvement. It includes collecting user feedback, analyzing usage data, and continuously refining the experience.

**Experience design** is a powerful tool for creating meaningful interactions that enhance user satisfaction, build brand loyalty, and drive innovation. By understanding the principles and processes of experience design, we can create products, services, and environments that enrich the lives of users and shape a more positive future.



## Lesson 12. Emotional Interaction

**Emotional interaction** is a complex and multifaceted concept that encompasses the exchange of emotional cues and responses between individuals or between humans and technology. It involves the expression, perception, and interpretation of emotions, and it plays a significant role in shaping our relationships, interactions, and overall experiences.

Key Elements of Emotional Interaction:

1. **Emotional Expression:** This involves the outward manifestation of emotions through verbal and nonverbal cues, such as facial expressions, body language, tone of voice, and word choice.
2. **Emotional Perception:** This involves the ability to recognize and interpret emotional cues from others. It requires attention to detail, empathy, and an understanding of nonverbal communication.
3. **Emotional Response:** This involves the internal and external reactions to perceived emotions. It can range from subtle changes in mood to more overt behaviors, such as expressing empathy, offering support, or withdrawing from interaction.



### **Emotional Interaction in Human-Computer Interaction (HCI):**

In the realm of HCI, emotional interaction has gained increasing attention due to its impact on user experience and the growing sophistication of artificial intelligence (AI) systems. Designers and researchers are exploring ways to incorporate emotional intelligence into technology, enabling machines to recognize, understand, and respond to human emotions.

#### Applications of Emotional Interaction in HCI:

1. **Adaptive User Interfaces:** Systems can adapt their behavior based on the user's emotional state, providing personalized support, adjusting the pace of interaction, or offering emotional support when needed.
2. **Enhanced Communication:** Emotional cues can be used to improve communication between humans and machines, enabling more natural and intuitive interactions.
3. **Empathetic AI Companions:** AI systems can act as empathetic companions, providing emotional support, understanding user needs, and offering guidance or assistance.
4. **Educational Applications:** Emotional interaction can enhance educational experiences, tailoring instruction to the student's emotional state and providing personalized feedback.

#### Challenges and Considerations in Emotional Interaction:

1. **Cultural and Individual Differences:** Emotional expression and interpretation vary across cultures and individuals, requiring careful consideration of context and cultural norms.
2. **Ethical Implications:** The use of emotional data and AI systems raises ethical concerns about privacy, manipulation, and potential biases.
3. **Technical Limitations:** Current technology is still limited in its ability to accurately detect and interpret emotions, making it challenging to develop fully reliable emotional interaction systems.

#### Future Directions of Emotional Interaction:



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubagui.edu](http://www.ubagui.edu)

E-mail Address: [sit@e.ubagui.edu](mailto:sit@e.ubagui.edu)

As AI and HCI continue to evolve, emotional interaction is poised to play an increasingly important role in shaping our interactions with technology. Future advancements may include:

1. More sophisticated emotion recognition and interpretation algorithms;
2. Development of AI systems that can generate and express emotions in a natural and nuanced way;
3. Integration of emotional interaction into a wider range of products, services, and environments;
4. Enhanced understanding of the ethical implications of emotional interaction and the development of appropriate safeguards;

Emotional interaction holds immense potential for enhancing our relationships with technology, creating more personalized, engaging, and supportive experiences. As we continue to explore and refine this field, we can unlock new possibilities for human-computer interaction and enrich the way we interact with the digital world.

**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubagui.edu](http://www.ubagui.edu)

E-mail Address: [sit@e.ubagui.edu](mailto:sit@e.ubagui.edu)

## FINALS

### Lesson 13. Introduction to User Interfaces

#### Introduction to User Interfaces

In today's technology-driven world, user interfaces (UIs) have become ubiquitous, shaping how we interact with computers, smartphones, appliances, and countless other devices. A well-designed UI is essential for creating a positive user experience,



ensuring that users can easily understand and navigate the system to accomplish their goals.

### What is a User Interface?

A **user interface** (UI) is the point of interaction between a user and a product or service. It encompasses all the visual elements, including the layout, colors, typography, and graphics, as well as the interactive components, such as buttons, menus, and input fields. The UI acts as a bridge between the user's intentions and the system's capabilities, allowing users to control the system and receive feedback.

### Goals of User Interface Design

The primary goal of UI design is to create an interface that is:

- **User-centered:** The design should prioritize the needs, wants, and abilities of the target users.
- **Usable:** Users should be able to easily learn, understand, and use the interface without frustration or error.
- **Effective:** The interface should enable users to accomplish their tasks efficiently and accurately.
- **Enjoyable:** The interface should be aesthetically pleasing and engaging, creating a positive user experience.

### Key Elements of User Interface Design

1. **Visual Hierarchy:** The arrangement of visual elements should guide the user's attention to the most important information.
2. **Balance and Proportion:** The arrangement of elements should create a sense of visual harmony and equilibrium.



3. **Contrast:** Different elements should be distinguishable through variations in color, size, or weight to enhance readability and clarity.
4. **Unity:** The design should convey a consistent visual style, creating a cohesive and unified experience.
5. **Typography:** The selection and use of fonts should be appropriate for the content and context, ensuring readability and legibility.
6. **Feedback:** The interface should provide clear and timely feedback to user actions, keeping them informed and engaged.
7. **Accessibility:** The design should consider the needs of users with disabilities, ensuring that the interface is accessible to all.

### Types of User Interfaces

1. **Graphical User Interfaces (GUIs):** GUIs rely on visual elements, such as icons, menus, and windows, to provide an intuitive and user-friendly interface.
2. **Command-Line Interfaces (CLIs):** CLIs rely on typed commands to interact with the system, often used by programmers and system administrators.
3. **Voice-Based Interfaces (VUIs):** VUIs use voice commands to control devices or access information, particularly popular in smart speakers and voice assistants.
4. **Gesture-Based Interfaces (GBIs):** GBIs rely on hand gestures or body movements to interact with the system, often used in virtual reality (VR) and augmented reality (AR) applications.

### Conclusion

**User interface design** is a complex and multifaceted field that encompasses a wide range of factors, from user-centered design principles to visual aesthetics and interactive elements. By understanding the principles and practices of UI design, we can create interfaces that are not only functional but also enjoyable, accessible, and effective for all users.

## Lesson 14. Natural UI

### Interface Basics



## What is Natural User Interface (NUI)?

A **natural user interface (NUI)** is a user interface that allows users to interact with a computer or device using natural methods such as speech, touch, or gesture. NUIs are designed to be more intuitive and user-friendly than traditional graphical user interfaces (GUIs), which rely on menus, icons, and other visual elements.

## Benefits of NUIs

NUIs offer several benefits over GUIs, including:

- **Increased usability:** NUIs can be easier to learn and use, especially for users who are not familiar with computers or technology.
- **Enhanced accessibility:** NUIs can be more accessible to users with disabilities, such as those who are blind or have limited mobility.
- **More natural interaction:** NUIs allow users to interact with computers in a more natural way, such as by speaking or gesturing.

## Types of NUIs

There are three main types of NUIs:

- **Speech recognition:** This type of NUI allows users to interact with computers using their voice. For example, users can use speech recognition to dictate text, control applications, or search for information.
- **Gesture recognition:** This type of NUI allows users to interact with computers using their hands or other body movements. For example, users can use gesture recognition to control games, navigate menus, or manipulate objects in virtual environments.
- **Touch recognition:** This type of NUI allows users to interact with computers using their touch. For example, users can use touch recognition to select items on a screen, zoom in and out on images, or swipe to navigate between different pages.



## Applications of NUIs

NUIs are used in a wide variety of applications, including:

- **Consumer electronics:** NUIs are becoming increasingly common in consumer electronics devices, such as smartphones, tablets, and smart speakers.
- **Automotive:** NUIs are used in cars to control infotainment systems, make hands-free calls, and avoid collisions.
- **Healthcare:** NUIs are used in healthcare to provide patients with more personalized and effective care.
- **Education:** NUIs are used in education to create more engaging and interactive learning experiences.

## Design Principles for NUIs

When designing NUIs, it is important to consider the following principles:

- **User-centered design:** NUIs should be designed to meet the needs of the target users.
- **Simplicity:** NUIs should be simple and easy to use.
- **Feedback:** NUIs should provide clear and timely feedback to user actions.
- **Accuracy:** NUIs should be accurate and reliable.
- **Accessibility:** NUIs should be accessible to users with disabilities.

## The Future of NUIs

**NUIs** are still in their early stages of development, but they have the potential to revolutionize the way we interact with computers. As NUI technology continues to improve, we can expect to see NUIs become even more common and sophisticated.



## LABORATORY

### Lesson 1: Webpage development with HTML

**HTML, or HyperText Markup Language**, is the foundation of web page development. It provides the structure and basic content for web pages, defining elements like headings, paragraphs, images, and links.

HTML (HyperText Markup Language) is the standard markup language for creating web pages. **It provides a set of codes that are used to define the structure and content of a web page**. HTML elements are the building blocks of HTML pages. They are used to define different parts of a web page, such as headings, paragraphs, images, and lists.

Here is a step-by-step guide on how to use HTML:

#### 1. Create an HTML document:

Open a text editor and create a new file. Save the file with an .html extension.

#### 2. Write the HTML declaration:



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

The first line of your HTML document should be the HTML declaration:

HTML

```
<!DOCTYPE html>
```

This tells the browser that the document is an HTML document.

3. Create the HTML structure:

The HTML structure consists of the `<html>`, `<head>`, and `<body>` tags:

HTML

```
<html>  
<head>  
<title>Your Page Title</title>  
</head>
```

```
<body >
```

Your page content will go here.

```
</body>  
</html>
```



The **<head>** tag contains information about the web page, such as **the title**. The **<body>** tag contains **the main content** of the web page.

#### 4. Add HTML elements:

HTML elements are used to define different parts of a web page. For example, the **<h1>** tag defines a heading, the **<p>** tag defines a paragraph, and the **<img>** tag defines an image.

To add an HTML element, simply type the tag name between angle brackets (<>). For example, to add a heading, you would type:

#### HTML

```
<h1>My Heading</h1>
```

#### 5. Close HTML tags:

Most HTML tags have a closing tag, which is the same as the opening tag but with a forward slash (/) at the beginning. For example, the closing tag for the **<h1>** tag is:

#### HTML

```
</h1>
```

#### 6. Save your HTML document:

Save your HTML document and open it in a web browser to see the results.

Here are some additional tips for using HTML:

- Use indentation to make your code easier to read and understand.



- Use comments to explain your code.
- Use a validator to check your HTML code for errors.

## Key Elements of HTML:

1. **HTML Tags:** Tags are the building blocks of HTML, enclosed in angle brackets (< >) and used to define different elements on the page. For example, the <h1> tag defines a heading, while the <p> tag defines a paragraph.
2. **Attributes:** Attributes provide additional information about HTML elements. They are specified within the opening tag and consist of a name-value pair separated by an equal sign (=). For instance, the <img> tag can have a src attribute to specify the image source.
3. **Document Structure:** HTML documents follow a hierarchical structure, starting with the <!DOCTYPE html> declaration and ending with the </html> closing tag. Within this structure, elements are nested to create the desired layout.
4. **Text Formatting:** HTML provides tags for formatting text, such as <b> for bold, <i> for italics, and <u> for underline. These tags can be nested within paragraphs or other text elements.
5. **Links (Hyperlinks):** The <a> tag creates hyperlinks, allowing users to navigate between web pages. It includes a href attribute to specify the destination URL.
6. **Images:** The <img> tag inserts images into a web page. It requires a **src** attribute to specify the image source and an alt attribute to provide alternative text for accessibility.
7. **Lists:** HTML offers various tags for creating lists, including <ul> for unordered lists, <ol> for ordered lists, and <li> for list items.
8. **Tables:** The <table> tag defines a table structure, while <tr> defines table rows, <th> table headers, and <td> table data cells.
9. **Forms:** HTML forms allow users to input data into a web page. They use elements like <form>, <input>, <select>, and <textarea> to create form fields and submit buttons.

## Example HTML Code:

HTML

```
<!DOCTYPE html>
```



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguo.edu](http://www.ubaguo.edu)

E-mail Address: [sit@e.ubaguo.edu](mailto:sit@e.ubaguo.edu)

```
<html>
<head>
<title>My Webpage</title>
</head>
<body>
<h1>Welcome to My Webpage</h1>
<p>This is a simple webpage created with HTML.</p>



<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>

<a href="https://www.example.com">Visit Another Webpage</a>
</body>
</html>
```



This basic example demonstrates the structure of an HTML document and the use of various HTML elements. For more complex web pages, HTML is often combined with CSS (Cascading Style Sheets) for styling and JavaScript for dynamic behavior.

## Lesson 2: Development environment

A **development environment** is a collection of software applications and tools that are used to develop software applications. It typically includes the following components:

- **Integrated Development Environment (IDE):** An IDE is a software application that provides a comprehensive set of tools for software development. It typically includes a source code editor, debugger, compiler, and other tools that are specific to the programming language being used.
- **Source Code Editor:** A source code editor is a software application that allows programmers to write and edit source code. It typically includes features such as syntax highlighting, code completion, and error checking.
- **Debugger:** A debugger is a software application that allows programmers to debug their code. It allows them to step through their code line by line, set breakpoints, and examine variables.
- **Compiler:** A compiler is a software application that translates source code into machine code. Machine code is the type of code that can be executed by a computer's processor.



Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

- **Build Automation Tools:** Build automation tools are software applications that automate the process of building software. They typically include tasks such as compiling, linking, and packaging the software.
- **Version Control System:** A version control system is a software application that allows programmers to track changes to their code. It allows them to revert to previous versions of their code if necessary.

In addition to these components, a development environment may also include other tools, such as a code profiler, a unit testing framework, and a continuous integration (CI) server.

The specific components of a development environment will vary depending on the type of software being developed and the programmer's preferences. However, all development environments should include the core components listed above.

Here are some of the **benefits of using a development environment:**

- **Increased productivity:** Development environments can help programmers to be more productive by providing them with a set of tools that can automate many of the tasks involved in software development.
- **Improved code quality:** Development environments can help programmers to write better code by providing them with tools that can check for syntax errors, logical errors, and other potential problems.
- **Reduced development time:** Development environments can help to reduce the time it takes to develop software by automating many of the tasks involved in the development process.
- **Improved collaboration:** Development environments can help programmers to collaborate more effectively by providing them with tools for sharing code and tracking changes.

Overall, development environments are an essential tool for software development. They can help programmers to be more productive, write better code, and reduce development time.



## Lesson 3: Basic Tags

**Basic HTML tags** are the fundamental building blocks of web pages. They provide the structure and basic content for web pages, defining elements like headings, paragraphs, images, and links.

Essential Basic HTML Tags:

1. <html> and </html>: These tags define the beginning and end of an HTML document.
2. <head> and </head>: These tags define the head section of the HTML document, where metadata and links are placed.
3. <body> and </body>: These tags define the body section of the HTML document, where the visible content of the web page is placed.
4. <h1> to <h6>: These tags define heading levels, from the largest (h1) to the smallest (h6).



5. <p>: This tag defines a paragraph of text.
6. : This tag inserts an image into the web page. The src attribute specifies the image source, and the alt attribute provides alternative text for accessibility.
7. <a href="https://www.example.com">Link text</a>: This tag creates a hyperlink, allowing users to navigate to another web page. The href attribute specifies the destination URL.
8. <ul> and </ul>: These tags define an unordered list.
9. <ol> and </ol>: These tags define an ordered list.
10. <li>: This tag defines a list item within an unordered or ordered list.
11. <table> and </table>: These tags define a table structure.
12. <tr>: This tag defines a table row.
13. <th>: This tag defines a table header cell.
14. <td>: This tag defines a table data cell.
15. <form>: This tag defines a form that allows users to submit data to the server.
16. <input type="text">, <input type="email">, <input type="password">: These tags create input fields for user input.
17. <textarea>: This tag creates a multi-line text input field.
18. <button type="submit">Submit</button>: This tag creates a button that submits the form data to the server.

These basic HTML tags provide the foundation for creating simple web pages. More complex web pages often involve combining HTML with CSS (Cascading Style Sheets) for styling and JavaScript for dynamic behavior.

Key elements within <head>:

- <title>: Specifies the title that appears in browser tabs and search engine results.
- <meta>: Provides metadata for search engines, browser compatibility, character encoding, and more.
- <link>: Links external resources like stylesheets (CSS files) and favicons.
- <script>: Includes JavaScript code within the header (though often placed in the <body>).

## Lesson 4: Formatting Tags



**Formatting tags in HTML** are used to enhance the appearance and organization of text on a web page. They allow you to control the style, size, and emphasis of text, making it more visually appealing and easier to read.

### Common Formatting Tags:

1. **<b>**: This tag makes the enclosed text bold.

Example: **<b>This is bold text.</b>**

2. **<i>**: This tag makes the enclosed text italic.

Example: *<i>This is italic text.</i>*

3. **<u>**: This tag underlines the enclosed text.

Example: <u>This is underlined text.</u>

4. **<sup>**: This tag superscripts the enclosed text, raising it slightly above the baseline.

Example: <sup><sup>This is superscripted text.</sup></sup>

5. **<sub>**: This tag subscripts the enclosed text, lowering it slightly below the baseline.

Example: <sub><sub>This is subscripted text.</sub></sub>

6. **<small>**: This tag makes the enclosed text appear smaller.

Example: <small>This is smaller text.</small>

7. **<mark>**: This tag highlights the enclosed text, drawing attention to it.



Example: <mark>This is highlighted text.</mark>

8. <del>: This tag marks the enclosed text as deleted, indicating that it is no longer valid or relevant.

Example: <del>This is deleted text.</del>

9. <ins>: This tag marks the enclosed text as inserted, indicating that it is new or updated information.

Example: <ins>This is inserted text.</ins>

### Using Formatting Tags Effectively:

1. **Use sparingly:** Avoid overusing formatting tags, as it can make the text look cluttered and difficult to read.
2. **Consider context:** Use formatting tags to emphasize important points or highlight key information.
3. **Maintain consistency:** Apply formatting tags consistently throughout the page to maintain a cohesive visual style.
4. **Balance aesthetics and accessibility:** Ensure that formatting tags do not impair the readability or accessibility of the text for all users.
5. **Combine with CSS:** Use CSS (Cascading Style Sheets) to control the overall styling of text, including font family, font size, and color.

## Lesson 5: Lists

**Lists** are an essential tool for organizing and presenting information in a clear and concise way. HTML provides two main types of lists: unordered lists and ordered lists.

### Unordered Lists (UL)



**Unordered lists** are used to present items in a non-sequential order. They are typically used for displaying lists of items that are not meant to be ranked or numbered.

The syntax for an unordered list is as follows:

HTML

```
<ul>  
<li>Item 1</li>  
<li>Item 2</li>  
<li>Item 3</li>  
</ul>
```

The **<ul> tag** defines the beginning of the unordered list, and the **</ul>** tag defines the end of the list. Each list item is defined using the **<li>** tag.

### Ordered Lists (OL)

**Ordered lists** are used to present items in a sequential order, typically numbered from 1 to the number of items in the list.

The syntax for an ordered list is as follows:

HTML

```
<ol>  
<li>Item 1</li>  
<li>Item 2</li>  
<li>Item 3</li>  
</ol>
```



The **<ol>** tag defines the beginning of the ordered list, and the **</ol>** tag defines the end of the list. Each list item is defined using the **<li>** tag. The **<ol>** tag can optionally include a type attribute to specify the numbering style, such as "1", "a", "A", "i", or "l".

## **Styling Lists with CSS**

**CSS (Cascading Style Sheets)** can be used to customize the appearance of lists, including the font, size, color, and spacing of list items. For example, the following CSS code would style unordered list items to be bold and green:

CSS

```
ul li {  
    font-weight: bold;  
    color: green;  
}
```

## **Nested Lists**

Lists can be nested within each other to create hierarchical structures. For example, the following HTML code creates a nested list with one ordered list containing two unordered lists:

HTML

```
<ol>  
<li>  
<ul>  
<li>Item 1</li>
```



<li>Item 2</li>

</ul>

</li>

<li>

<ul>

<li>Item 3</li>

<li>Item 4</li>

</ul>

</li>

</ol>

**Lists** are a versatile tool for presenting information in a clear and organized way. By using both unordered and ordered lists, and styling them with CSS, you can effectively communicate information to your audience.

## Lesson 6: Tables

**Tables** are a valuable tool for presenting tabular data in a structured and organized manner. HTML provides the necessary tags to create and define tables, allowing you to present data in a clear and concise way.

### Creating Tables in HTML



The **basic structure of an HTML table** involves the following tags:

1. <table>: This tag marks the beginning of the table.
2. </table>: This tag marks the end of the table.
3. <tr>: This tag defines a table row.
4. <th>: This tag defines a table header cell, typically used for column labels.
5. <td>: This tag defines a table data cell, typically used for displaying data within the table.

Here's an example of a simple HTML table:

HTML

```
<table>

<tr><th>Product</th><th>Price</th><th>Quantity</th></tr>

<tr><td>Laptop</td><td>$1,200</td><td>2</td></tr>

<tr><td>Smartphone</td><td>$500</td><td>3</td></tr>

</table>
```

## **Styling Tables with CSS**

**CSS (Cascading Style Sheets)** can be used to enhance the appearance and readability of tables. CSS allows you to control various aspects of the table's presentation, including:



- **Border styles:** Define the thickness, style, and color of table borders.
- **Cell padding and spacing:** Adjust the spacing between table cells and their contents.
- **Cell background and text styles:** Customize the background color, text color, and font styles for table cells.
- **Table alignment:** Align the table and its contents within the page.

## Advanced Table Features

HTML provides additional tags for more **complex table structures**:

1. **<caption>**: This tag defines a caption for the table, providing a brief description or summary of the table's contents.
2. **<colgroup>**: This tag defines a group of columns, allowing you to apply styling or attributes to multiple columns at once.
3. **<thead>**: This tag defines the table header section, typically containing column labels.
4. **<tfoot>**: This tag defines the table footer section, typically used for summary information or notes.
5. **<tbody>**: This tag defines the table body section, containing the main data rows of the table.

By combining these tags and leveraging CSS styling, you can create tables that are both informative and visually appealing. Tables play a crucial role in presenting data effectively and enhancing the overall user experience of web pages and applications.

## Lesson 7: Media

HTML provides tags for embedding media elements, such as images, videos, and audio files, into web pages. This allows you to enrich your web content with multimedia elements and enhance the user experience.

### Embedding Images in HTML



The `<img>` tag is used to insert images into web pages. It requires a `src` attribute to specify the image source URL and an `alt` attribute to provide alternative text for accessibility purposes.

Here's an example of **embedding an image**:

HTML

```

```

### **Embedding Videos in HTML**

The `<video>` tag is used to embed videos into web pages. It supports various video formats, including MP4, WebM, and Ogg. The `<video>` tag can include `src` attributes to specify the video source URLs, `controls` attribute to display playback controls, and other attributes to control video playback and styling.

Here's an example of **embedding a video**:HTML

```
<video src="video.mp4" controls>  
  <source  
    src="video.webm"  
    type="video/webm">  
  
<source
```



Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

**src="video.ogv"**

**type="video/ogg">**

Your browser does not support the video tag.

**</video>**

## Embedding Audio Files in HTML

The **<audio>** tag is used to embed audio files into web pages. It supports various audio formats, including MP3, WAV, and OGG. The **<audio>** tag can include **src** attributes to specify the audio source URLs, **controls** attribute to display playback controls, and other attributes to control audio playback and styling.

Here's an example of **embedding an audio file**:

HTML

```
<audio src="audio.mp3" controls>  
  <source src="audio.wav" type="audio/wav">  
  <source src="audio.ogv" type="audio/ogg">
```

Your browser does not support the audio tag.

**</audio>**

By embedding media elements into your web pages, you can enhance the user experience and provide a more engaging and interactive experience for your audience.

1. [github.com/sparkbox/bouncy-ball](https://github.com/sparkbox/bouncy-ball)

## Lesson 8: Links



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

**Links** are essential for providing navigation and allowing users to explore different parts of a web page or connect to other web pages. HTML provides the `<a>` tag to create hyperlinks.

## Syntax for Links

The basic syntax for a link is as follows:

HTML

```
<a href="url">link text</a>
```

The `<a>` tag defines a hyperlink, the **href attribute** specifies the URL of the destination page, and the **link text** is the visible text that users will click to navigate to the linked page.

Here's an example of a link to a Google search:

HTML

```
<a href="https://www.google.com">Search Google</a>
```

## Target Attributes

The **target attribute** of the `<a>` tag specifies where to open the linked page. The default target is the current window, but you can use `target="_blank"` to open the linked page in a new browser tab.

Here's an example of a link that opens a new tab:

HTML

```
<a href="https://www.example.com" target="_blank">Open in New Tab</a>
```

## Rel and Rev Attributes



The **rel attribute** of the <a> tag specifies the relationship between the current page and the linked page. Common values for rel include nofollow (indicates that search engines should not follow the link) and noopener noreferrer (prevents the linked page from opening the current page in a new tab and accessing its cookies).

The **rev attribute** of the <a> tag specifies the reverse relationship between the current page and the linked page. This is typically used for reciprocal linking or link exchange agreements.

## Styling Links

**CSS (Cascading Style Sheets)** can be used to customize the appearance of links. For example, you can change the color, font, and underline style of links.

Here's an example of styling links to use a **different color and font**:

CSS

```
a {  
    color: blue;  
    font-family: sans-serif;  
    font-size: 16px;  
    text-decoration: none;  
}
```

**Links** are a fundamental element of web pages, providing navigation, connecting to external resources, and enhancing user experience. By understanding the syntax and attributes of links, you can create effective links that guide users through your website and connect them to relevant information.



## Lesson 9 : Forms

**Forms** are an essential tool for collecting user input on web pages. HTML provides various tags for creating forms, allowing you to gather information from users, submit data to a server, and perform various actions based on user input.

### Creating Forms with HTML

The **<form>** tag defines the beginning of a form and the **</form>** tag defines the end of the form. Within the form, you can use various input tags to collect user data. Common input tags include:

- **<input type="text">**: Creates a text input field for user input.
- **<input type="email">**: Creates an email input field with email validation.
- **<input type="password">**: Creates a password input field with password masking.
- **<textarea>**: Creates a multi-line text input field for longer text input.
- **<select>**: Creates a dropdown menu with multiple options for user selection.
- **<checkbox>**: Creates a checkbox input for selecting or deselecting an option.
- **<radio>**: Creates a radio button input for selecting one option among a group.
- **<submit>**: Creates a submit button to submit the form data to a server.

### Styling Forms with CSS

CSS (Cascading Style Sheets) can be used to customize the appearance of forms, including the layout, styling of input fields, and button design.

### Handling Form Submission

The action attribute of the **<form>** tag specifies the URL where the form data will be submitted. The method attribute specifies the HTTP method used for form submission, typically either POST or GET.



When a user submits a form, the form data is sent to the specified URL, and the server can process the data and perform actions based on the user input.

## Form Validation

HTML provides some basic form validation with the required attribute, which can be applied to input fields to indicate that they must be filled in before submitting the form. However, for more complex validation rules, JavaScript is often used to validate user input and ensure data integrity.

## Common Use Cases for Forms

Forms are used for a wide variety of purposes on web pages, including:

- **User registration and login:** Collecting user information for account creation and authentication.
- **Contact forms and feedback forms:** Gathering user feedback, inquiries, or support requests.
- **Product order forms and shopping carts:** Processing online purchases and collecting payment information.
- **Surveys and questionnaires:** Gathering user opinions, preferences, or demographic data.
- **Search forms and filters:** Enabling users to refine their search criteria and filter results.

Forms play a crucial role in web applications, providing a structured and interactive way to collect user input, process data, and enhance user engagement. By understanding the principles of form creation, validation, and submission, you can effectively integrate forms into your web pages and applications.



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

## MIDTERMS

### Lesson 10: Introduction to CSS

Certainly! **CSS (Cascading Style Sheets)** is a style sheet language used to style HTML elements. It allows you to control the appearance of your web pages, including the layout, colors, fonts, and other visual elements.

Key Features of CSS:

Benefits of Using CSS:



Feature	Description
Separation of concerns	CSS separates the presentation of content from the content itself, making it easier to maintain and update web pages.
Selectors	CSS uses selectors to identify the HTML elements that you want to style. Selectors can be based on element type, class, ID, attribute, or a combination of these.
Properties and values	CSS properties define the style attributes that you want to apply to the selected elements. Properties have corresponding values that specify the desired style.



Benefit	Description
Inheritance	CSS properties are inherited by child elements from their parent elements, unless overridden by specific styles.
Specificity	The specificity of a CSS selector determines which style rule takes precedence when multiple rules apply to the same element.
Improved visual appeal	CSS enhances the visual appearance of web pages, making them more engaging and user-friendly.
Consistent styling	CSS ensures consistent styling across different pages and browsers.
Accessibility	CSS can be used to improve the accessibility of web pages for users with disabilities.
Search engine optimization (SEO)	CSS can indirectly impact SEO by improving the overall user experience and page load times.
Learning CSS:	



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

CSS is a relatively easy language to learn, especially if you have some experience with HTML. There are numerous resources available online and in libraries to help you learn CSS, including tutorials, documentation, and interactive courses.

Conclusion:

CSS is an essential tool for web development, allowing you to create visually appealing, user-friendly, and accessible web pages. By learning CSS, you can enhance the overall quality and effectiveness of your web projects.

## Lesson 11: CSS properties and attributes

**CSS properties and attributes** are the building blocks of CSS styling. They work together to define the style of HTML elements on a web page.

CSS Properties:



**CSS properties** are the fundamental elements of CSS styling. They define the specific style attributes that you want to apply to HTML elements. Each property has a corresponding name and value. For instance, the color property sets the text color of an element, and its value can be a color name (e.g., red, blue, green) or a hexadecimal color code (e.g., #FF0000, #0000FF, #00FF00).

CSS Attributes:

**CSS attributes** are modifiers of CSS properties. They provide additional information or context to properties, allowing for more refined control over the styling. Attributes are typically used within property values, often enclosed in parentheses. For example, the border-radius property sets the roundedness of an element's borders, and the attribute 5px specifies the desired radius in pixels.

Distinction between Properties and Attributes:

The key distinction between properties and attributes lies in their purpose and usage:

- **Properties:** Properties define the primary style attributes, such as color, font size, and background color.
- **Attributes:** Attributes modify or enhance properties, providing additional details or context, such as border radius, padding, and margin.

Examples of Properties and Attributes:



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

Property	Attribute	Description
color	#FF0000	Sets the text color to red
font-family	Arial, Helvetica, sans-serif	Sets the font family to Arial, Helvetica, or a similar sans-serif font
font-size	16px	Sets the font size to 16 pixels
background-color	#00FF00	Sets the background color to green
border-width	5px	Sets the width of the border to 5 pixels
border-style	solid	Sets the border style to solid



border-color	#0000FF	Sets the border color to blue
or		
padding	10px 20px	Sets the padding around the element's content to 10 pixels top and bottom, and 20 pixels left and right
margin	0 auto	Sets the margin around the element to 0 pixels top, bottom, and left, and sets the right margin to auto, automatically centering the element horizontally

CSS properties and attributes work together to create a comprehensive set of tools for styling HTML elements and enhancing the appearance of web pages. Understanding the distinction between properties and attributes is crucial for effectively controlling the visual elements of web pages using CSS.



## Lesson 12: CSS implementation

CSS can be implemented in three primary ways:

1. **Inline CSS:** Inline CSS is applied directly to HTML elements using the style attribute. This method is suitable for applying quick and simple styles to individual elements.

Example:

HTML

```
<p style="color: red; font-size: 20px;">This text is red and 20px in size.</p>
```

2. **Internal CSS:** Internal CSS is placed within the <head> section of an HTML document using the <style> tag. This method is useful for styling multiple elements within a single page.

Example:

HTML

```
<head>  
<style>  
p {  
color: blue;
```



```
font-size: 16px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This text is blue and 16px in size.</p>
```

```
<p>This text is also blue and 16px in size.</p>
```

```
</body>
```

3. **External CSS:** External CSS is stored in a separate CSS file and linked to the HTML document using the `<link>` tag in the `<head>` section. This method is preferred for large and complex stylesheets, as it promotes code reusability and separation of concerns.

Example:

HTML

```
<head>  
<link rel="stylesheet" href="style.css">  
</head>
```

```
<body>
```

```
</body>
```



In addition to these three main methods, CSS can also be implemented using CSS frameworks like Bootstrap or Foundation, which provide pre-built components and styles, simplifying the development process.

The choice of CSS implementation method depends on the scope of the project, the complexity of the styles, and the developer's preference. Inline CSS is suitable for quick changes, internal CSS is useful for small to medium projects, and external CSS is preferred for large projects and code maintainability.

## Lesson 13: Responsive Web page design

**Responsive web page design (RWD)** is an approach to web design that ensures that web pages adapt their layout to different screen sizes and devices. This means that websites designed with RWD will look good and function properly on desktops, laptops, tablets, smartphones, and other devices.

### Key Principles of Responsive Web Design:

1. **Fluid Grid:** RWD utilizes a fluid grid system, where the width of elements is defined as percentages rather than fixed pixels. This allows elements to resize proportionally as the viewport changes.
2. **Media Queries:** Media queries are CSS techniques that allow you to target specific device types or screen sizes and apply different styles accordingly. This ensures that the website's layout and presentation adapt to the device's capabilities.
3. **Adaptive Images:** Responsive images automatically adjust their size and resolution based on the device's screen size and resolution, ensuring optimal image quality and performance.
4. **Progressive Enhancement:** RWD follows the principle of progressive enhancement, providing a basic level of functionality for all devices and



progressively enhancing the experience for devices with more advanced capabilities.

### **Benefits of Responsive Web Design:**

1. **Improved User Experience:** RWD ensures a consistent and positive user experience across different devices, making it easier for users to access and interact with the website.
2. **Increased Website Reach:** By catering to a wider range of devices, RWD increases the website's potential audience and accessibility.
3. **Reduced Maintenance:** RWD simplifies website maintenance by eliminating the need for separate mobile websites or dedicated mobile development efforts.
4. **Search Engine Optimization (SEO) Benefits:** Responsive websites are generally considered more favorably by search engines, potentially boosting SEO performance.

### **Implementing Responsive Web Design:**

1. **Choose a Responsive Framework:** Several CSS frameworks, such as Bootstrap, Foundation, and Materialize, provide pre-built responsive components and layouts, making RWD implementation easier.
2. **Establish a Mobile-First Approach:** Start by designing for the smallest viewport first, ensuring a seamless experience on mobile devices and gradually adapting for larger screens.
3. **Utilize Media Queries:** Employ media queries to adjust styles and layouts based on specific device types or screen sizes.
4. **Optimize Images:** Use adaptive images to ensure optimal image quality and performance across devices.
5. **Thorough Testing:** Test the website extensively on various devices and screen sizes to ensure consistent performance and responsiveness.

Responsive web design has become an essential aspect of modern web development, ensuring that websites provide an optimal user experience across the diverse range of



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguo.edu](http://www.ubaguo.edu)

E-mail Address: [sit@e.ubaguo.edu](mailto:sit@e.ubaguo.edu)

devices used today. By embracing RWD principles, developers can create websites that are accessible, engaging, and effective in reaching their target audience.

---

**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguo.edu](http://www.ubaguo.edu)

E-mail Address: [sit@e.ubaguo.edu](mailto:sit@e.ubaguo.edu)

## FINALS

### Lesson 14: Introduction to JavaScript

**JavaScript** is a scripting language that is commonly used to add interactivity and dynamic behavior to web pages. It is a versatile language that can also be used for server-side programming, game development, and other applications.



## Variables

**Variables** are used to store data in JavaScript. They are declared using the var, let, or const keyword, followed by the variable name and an optional initial value.

### JavaScript

```
// Using 'var' keyword
```

```
var age = 30;
```

```
// Using 'let' keyword
```

```
let name = "John Doe";
```

```
// Using 'const' keyword for constants
```

```
const PI = 3.14159;
```

## Data Types

JavaScript has various data types to represent different kinds of information. Common data types include:

- Numbers: Represent numerical values, including integers (e.g., 1, 2, 3) and decimals (e.g., 10.5, 3.14).
- Strings: Represent sequences of characters, enclosed in single or double quotes (e.g., "Hello, World!", 'JavaScript').



- Booleans: Represent logical values, either true or false.
- Undefined: Represents the absence of a value.
- Null: Represents the intentional absence of a value.
- Objects: Represent collections of key-value pairs, where keys are properties and values can be of any data type.
- Arrays: Represent ordered collections of values, where each value has an index.

## JavaScript Functions

**Functions** are blocks of reusable code that perform specific tasks. They are declared using the function keyword, followed by the function name, parentheses, and an optional curly brace block containing the function's code.

JavaScript

```
function greet(name) {  
    console.log("Hello, " + name + "!");  
}
```

```
greet("Alice"); // Output: Hello, Alice!
```

**Functions** can take parameters, which are like placeholders for values that will be passed to the function when it is called. They can also return a value using the return keyword.



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

## JavaScript

```
function calculateArea(width, height) {  
    return width * height;  
}  
  
let area = calculateArea(10, 5); // area = 50
```

Functions play a crucial role in JavaScript programming, allowing you to modularize your code, promote reusability, and encapsulate complex operations.



## Lesson 15: Project design and development

**Project design and development** is the process of transforming a project idea into a tangible and functional product or service. It involves a series of steps that encompass planning, design, implementation, testing, and deployment.

### Project Design

**The project design phase** lays the foundation for the entire project lifecycle. It involves defining the project scope, objectives, deliverables, and milestones. It also involves identifying the project team, establishing project management tools, and creating a project plan.

Key aspects of project design include:

1. **Problem Definition:** Clearly articulate the problem or challenge that the project aims to address.
2. **Goal Setting:** Establish clear and measurable goals that the project will achieve.
3. **Scope Definition:** Define the boundaries of the project, including what will be included and excluded.
4. **Requirements Gathering:** Gather and analyze requirements from stakeholders to understand their needs and expectations.
5. **Design Alternatives:** Explore and evaluate potential solutions and approaches to address the problem.
6. **Solution Selection:** Choose the most feasible and effective solution based on evaluation criteria.
7. **Project Planning:** Create a detailed project plan that outlines tasks, timelines, resources, and budgets.



## Project Development

**The project development phase** involves translating the project design into a tangible product or service. It encompasses implementing the chosen solution, testing the product or service, and preparing for deployment.

Key aspects of project development include:

1. **Implementation:** Put the project plan into action, executing the tasks and activities outlined in the plan.
2. **Software Development:** Develop the software code or application that forms the core of the project.
3. **Testing:** Conduct thorough testing to ensure the product or service meets the requirements and functions as intended.
4. **Quality Assurance:** Implement quality assurance processes to identify and address any defects or bugs.
5. **Documentation:** Create comprehensive documentation that explains the product or service, its features, and its usage.
6. **Deployment:** Prepare the product or service for release and deployment to the target users or market.



## Project Management

**Project management** is an essential aspect of project design and development. It involves overseeing the project's progress, ensuring it stays on track to meet its goals and objectives within the defined timeline and budget.

Key aspects of project management include:

1. **Risk Management:** Identify potential risks that could impact the project and develop mitigation strategies.
2. **Change Management:** Manage project changes in a controlled manner to minimize disruption and ensure alignment with project objectives.
3. **Communication:** Maintain clear and effective communication among stakeholders, team members, and external parties.
4. **Stakeholder Management:** Engage with stakeholders, address their concerns, and manage expectations throughout the project lifecycle.
5. **Performance Monitoring:** Continuously monitor project progress, track key performance indicators, and identify areas for improvement.
6. **Project Closure:** Formally conclude the project, evaluating its success, documenting lessons learned, and archiving project materials.



Project Name: To-Do List Manager

**Problem Definition:** Many people struggle to manage their tasks effectively, leading to missed deadlines, reduced productivity, and increased stress.

**Goal:** Develop a user-friendly web application that allows users to create, manage, and track their tasks effectively, improving their task management skills and overall productivity.

**Scope Definition:** The project will focus on developing the core features of a to-do list application, including:

- Task creation, editing, and deletion
- Task prioritization and categorization
- Task completion tracking
- Due date reminders
- Simple search and filtering functionality

**Requirements Gathering:**

- Conduct user interviews and surveys to gather user needs and expectations.
- Analyze existing to-do list applications to identify best practices and features.

**Design Alternatives:**



Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

- Explore different UI/UX designs for the to-do list interface, focusing on usability and accessibility.
- Consider implementing various task management methodologies, such as Kanban or time blocking.

#### Solution Selection:

- Evaluate design alternatives based on user feedback, feasibility, and alignment with project goals.
- Select the design that best addresses user needs and provides a user-friendly and efficient task management experience.

#### Project Planning:

- Create a detailed project plan outlining tasks, dependencies, timelines, and resource allocation.
- Break down the project into manageable phases and milestones to track progress effectively.

#### Implementation:

- Develop the front-end of the web application using HTML, CSS, and JavaScript.
- Develop the back-end of the application using a suitable server-side programming language and database.
- Implement user authentication and authorization mechanisms to secure user data.

#### Testing:

- Conduct thorough unit testing to ensure the individual components of the application function correctly.
- Perform integration testing to verify the interaction between different components of the application.
- Conduct user acceptance testing to gather feedback from real users and identify any usability issues.



**Deployment:**

- Deploy the web application to a hosting platform, ensuring scalability and performance.
- Provide clear documentation and support resources for users to access.

**Project Closure:**

- Evaluate the project's success based on predefined criteria, such as user satisfaction, feature completeness, and budget adherence.
- Document lessons learned and identify areas for improvement for future projects.
- Archive project documentation and source code for future reference.

This example illustrates the iterative and structured approach to project design and development, ensuring that the project is well-defined, well-managed, and delivers a valuable product or service that meets user needs.

```
<!DOCTYPE html>

<html
lang="en">

<head>

<meta
charset="UTF-8">

<meta
name="viewport"
content="width=device-width, initial-scale=1.0">
```



Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

```
<title>Responsive Interface</title>
```

```
<style>
```

```
body {  
    font-family: sans-serif;  
    margin: 0;  
    padding: 0;  
}  
  
.container {  
    max-width: 800px;  
    margin: 0 auto;  
    padding: 20px;  
    background-color: #f2f2f2;  
}
```

```
.header {  
    text-align: center;  
    margin-bottom: 20px;  
}
```

```
.header h1 {  
    font-size: 24px;  
    font-weight: bold;
```



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)

E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

}

```
.main {  
  
    display: flex;  
  
    flex-wrap: wrap;  
  
    justify-content: space-between;  
  
}
```

```
.main .box {  
  
    width: 25%;  
  
    padding: 20px;  
  
    border: 1px solid #ccc;  
  
    margin-bottom: 20px;  
  
}
```

```
@media (max-width: 768px) {  
  
    .main .box {  
  
        width: 50%;  
  
    }  
  
}  
  
@media (max-width: 480px) {
```



**UNIVERSITY OF  
Baguio**  
**SCHOOL OF INFORMATION TECHNOLOGY**  
General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071      Website: [www.ubaguiro.edu](http://www.ubaguiro.edu)      E-mail Address: [sit@e.ubaguiro.edu](mailto:sit@e.ubaguiro.edu)

```
.main .box {  
    width: 100%;  
}  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<div class="container">  
  
<div class="header">  
  
<h1>Responsive Interface</h1>  
  
</div>  
  
<div class="main">  
  
<div class="box">  
  
<h2>Box 1</h2>  
  
<p>This is the content of Box 1.</p>  
  
</div>  
  
<div class="box">  
  
<h2>Box 2</h2>  
  
<p>This is the content of Box 2.</p>  
  
</div>  
  
<div class="box">
```



```
<h2>Box 3</h2>

<p>This is the content of Box 3.</p>

</div>

<div class="box">

<h2>Box 4</h2>

<p>This is the content of Box 4.</p>

</div>

</div>

</body>

</html>
```

This code will create a simple web page with four boxes arranged in a grid. The boxes will resize and rearrange themselves based on the width of the viewport, ensuring that the interface looks good on different devices.

Here is a breakdown of the code:

- The `<html>` tag defines the root element of the HTML document.
- The `<head>` tag contains metadata about the document, such as the character encoding and viewport configuration.
- The `<body>` tag contains the main content of the web page.
- The `<div>` tag is used to create reusable blocks of content. In this case, it is used to create the container, header, main, and boxes.
- The `<h1>` tag defines a heading element, which is used to create the title "Responsive Interface".



Telefax No.: (074) 442-3071

Website: [www.ubaguiio.edu](http://www.ubaguiio.edu)

E-mail Address: [sit@e.ubaguiio.edu](mailto:sit@e.ubaguiio.edu)

- The <p> tag defines a paragraph element, which is used to create the text content of the boxes.
- The @media rule is used to specify styles that should only apply to specific media types, such as screen size. In this case, it is used to adjust the width of the boxes based on the screen size.

**\*Create an Individual Final Project\***