U23MEO11/ APPLIED DESIGN THINKING

UNIT I

DESIGN THINKING PRINCIPLES

Period 1

Exploring Human-Centered Design

1.Introduction

"Today, technology is advancing very fast.
We have AI, automation, smart devices.
Still, many products fail.
Why do you think this happens?"

Most failures happen because:

- Designers think **what is best**
- But users think **what is comfortable**

This gap leads us to **Human-Centered Design**.

2.Meaning of Human-Centered Design (HCD)

Definition:

Human-Centered Design is a design approach that prioritizes the needs, experiences, emotions, and limitations of people at every stage of the design process.

Very Simple Definition:

Human-Centered Design means:

- First understand people
- Then understand their problems
- Then design solutions

Daily Life Example:

If you are buying shoes:

- You don’t choose the most advanced shoe
- You choose the one that fits your foot comfortably

3.Why Human-Centered Design is Necessary

Every person is different:

- Age
- Physical ability
- Education
- Environment

A product designed without considering this will:

- Confuse users
- Frustrate users
- Be rejected

Important Point:

“A technically perfect product can fail if it is difficult to use.”

Example:

A government website with:

- Complex language
- Too many steps

Even though it works technically, people avoid using it.

4.Traditional Design vs Human-Centered Design

Traditional Design:

- Starts with technology
- Designer assumes user needs
- Testing happens at the end
- Changes are expensive

Human-Centered Design:

- Starts with people
- Users are studied deeply
- Testing happens early
- Improvements are continuous

"Human-Centered Design reduces risk by finding mistakes early."

5.Core Principles of Human-Centered Design

Principle 1: Empathy – The Heart of HCD

What is Empathy?

Empathy means:

- Seeing the problem through the user's eyes
- Feeling what the user feels
- Understanding daily struggles

Difference between Sympathy and Empathy

- Sympathy: Feeling sorry for someone
- Empathy: Understanding their experience

Example:

Designing a mobile app for elderly users:

- Small text = difficult
- Complex navigation = confusing

A designer with empathy:

- Uses large fonts
- Simple screens

Principle 2: User Involvement – Users are Partners

In Human-Centered Design:

- Users are not outsiders
- They are part of the design team

User involvement includes:

- Interviews
- Feedback sessions
- Prototype testing

> "Users help designers avoid wrong assumptions."

Principle 3: Iteration – Learning by Doing

Design Thinking believes:

- First solution is never perfect
- Improvement comes from testing

Iteration cycle:
Design → Test → Learn → Improve

"Failure is a learning step, not a final result."

Principle 4: Multidisciplinary Approach – Many Minds, One Solution

Human problems are complex.

To solve them, we need:

- Engineers → feasibility
- Designers → usability
- Psychologists → behaviour
- Business → viability

This creates **balanced innovation**.

6.Real-Life Case Example – ATM Design

Step 1: Problem

Early ATMs were:

- English only
- Complex
- Confusing

Step 2: User Problems

- Elderly users struggled
- Rural users avoided ATMs

Step 3: Human-Centered Changes

- Regional languages
- Voice instructions
- Simple User Interface (UI)

Step 4: Result

- Increased usage
- Better satisfaction

"Design improved after understanding users."

7.Classroom Activity – Observation & Thinking

Activity Title: *Identify a Poor Design*

Steps:

1. Ask students to think of any daily-use product
2. Identify one difficulty
3. Suggest improvement

This activity builds:

- Empathy
- Observation
- Critical thinking

8.Common Design Mistakes

☐ Designing based on assumptions
☐ Ignoring user emotions
☐ Over-engineering solutions

Human-Centered Design avoids all these.

9.Key Takeaways

- HCD puts humans first
- Empathy leads to better solutions
- Users reduce risk

- Iteration improves quality
- Better design = better adoption

Period 2:

Understanding the Innovation Process

1.What is Innovation?

Definition:

Innovation is the process of converting ideas into **solutions that create value and are accepted by users.**

- **Idea alone is not innovation**
- Innovation must:
  - ✓ Solve a real problem
  - ✓ Create value
  - ✓ Be usable and adopted

Innovation = Idea + Value + Implementation

- **Idea** → Thought or concept
- **Value** → Benefit to users
- **Implementation** → Practical execution

2. Difference Between Invention and Innovation

Invention	Innovation
New idea or device	Applying idea to solve problems
May not be used	Must be accepted by users
Technology-focused	User-focused

Example:

- Invention: Touch screen technology
- Innovation: Smartphones with intuitive apps

"Innovation succeeds only when people use it."

3.Why Do We Need an Innovation Process?

Without a process:

- Ideas remain random
- High failure risk
- Wasted time and money

With a process:

- Structured thinking
- Reduced risk
- Better outcomes

4.Innovation Process in Design Thinking

Step 1: Problem Identification

- Innovation starts with **problems, not solutions**
- Problems are identified through:
  - ✓ Observation
  - ✓ Interviews
  - ✓ Empathy

Example:
Students struggling with attendance marking → Problem identified.

Step 2: Ideation

- Generating many ideas
- No judgment initially
- Creativity encouraged

"More ideas increase the chance of a good solution."

Step 3: Prototyping

- Creating simple models
- Low-cost
- Quick to build

Types:

- Paper sketches
- Mock-ups
- Digital wireframes

"Prototype is not the final product."

Step 4: Testing and Validation

- Testing with real users
- Gathering feedback
- Learning what works and what doesn't

"Testing saves cost by detecting mistakes early."

Step 5: Iteration and Improvement

- Improve based on feedback
- Repeat the cycle

"Innovation is continuous."

5. Real-Life Example – Innovation Process

Example: Online Food Delivery Apps

Step 1: Problem
People don't want to cook or travel to restaurants.

Step 2: Ideation
Apps, phone calls, delivery services.

Step 3: Prototyping
Basic app with limited features.

Step 4: Testing
User feedback on speed, UI, payment.

Step 5: Iteration
Added tracking, reviews, discounts.

"Success came through continuous improvement."

6.Classroom Activity

Activity: Problem to Idea

Steps:

1. Ask students to identify one daily problem
2. Suggest two solutions
3. Discuss feasibility

Purpose:

- Understand problem-first approach
- Encourage creativity

7.Common Mistakes in Innovation

☐ Jumping to solutions too early

☐ Ignoring user feedback

☐ No validation

"A good process prevents these mistakes."

8. Key Takeaways

- Innovation is systematic
- Problems come before solutions
- Prototyping reduces risk
- Testing ensures acceptance
- Iteration leads to success

Period 3

Discovering Areas of Opportunity

1.What is an Opportunity Area?

Definition:

An **Opportunity Area** is a situation where:

- Users face difficulties
- Needs are unmet
- Existing solutions are inefficient or absent

Simple words:

"An opportunity area is a chance to improve people's lives by solving a real problem."

Simple Analogy:

Think of a **problem as a signal** that something can be improved.

- **Long queue**
  People waste time standing in line.
  This shows an **opportunity to create a better waiting system** (online booking, token system).
- **Confusing mobile app**
  Users cannot easily find buttons or features.
  This shows an **opportunity to redesign the app** so it is simple and easy to use.
- **Slow service**
  Work takes too much time and frustrates users.
  This shows an **opportunity to make the process faster and more efficient**.

Conclusion:

Problems are **not failures**.
They show **where improvement is needed**.

Problems = Opportunities for designers to innovate and improve people

2. Difference Between Problem and Opportunity

- **Problem** → Something that causes difficulty
- **Opportunity** → A possibility to create value by solving that problem

Example:

- Problem: Students forget assignment deadlines
- Opportunity: Reminder or task management app

Design thinkers see problems as opportunities.

3.Why Discovering Opportunity Areas is Important?

If opportunity is not identified correctly:

- Solution may not be useful
- Users may not care
- Innovation may fail

Correct opportunity identification helps:

- Focus efforts
- Reduce risk
- Increase impact

"Innovation fails more due to wrong problem selection than poor solutions."

4.Sources of Opportunity Areas

1. User Pain Points

- Pain points are frustrations faced by users
- Found through observation and interviews

Example:

- Difficulty in online payment
- Poor public transport information

2. Inefficiencies in Existing Systems

- Processes that are slow, complex, or outdated

Example:

- Manual attendance system
- Paper-based hospital records

3. Behavioral Observations

- Observe what users do, not what they say
- Users adapt to bad systems silently

Example:

- People standing instead of sitting → poor seating design

4. Technological Changes

- New technology creates new opportunities

Example:

- Smartphones → mobile banking apps

5. Social and Environmental Issues

- Problems related to society and environment are strong opportunity areas

Example:

- Waste management
- Water scarcity

5.Methods to Discover Opportunity Areas

❖ **Observation**

- Watch users in real environments

- Identify difficulties and workarounds

Example:

- Observing students during registration process

❖ **Interviews**

- Ask open-ended questions
- Understand emotions and experiences

Example:

- "Tell me about the problems you face while booking tickets."

❖ **Journey Mapping**

- Map the user's step-by-step experience
- Identify pain points at each step

❖ **Empathy Mapping**

- Understand what the user:
  - ✓ Says
  - ✓ Thinks
  - ✓ Does
  - ✓ Feels

6. Real-Life Example

Example: Hospital Waiting Time

Observation:

- Long queues
- Confused patients

User Pain:

- Anxiety
- Time wasted

Opportunity Area:

- Digital appointment system
- Queue display screens

Explain:

“Innovation opportunity came from observing real pain.”

7.Classroom Activity

Activity: Spot the Opportunity

Steps:

1. Ask students to observe one problem in the college
2. Identify:
    - ✓ Who is affected?
    - ✓ What is the pain?
    - ✓ Why is it important?
3. Convert it into an opportunity statement

Example:

“There is an opportunity to improve _____ for _____ by _____.”

Explain purpose:

- Builds observation skills
- Encourages problem-first thinking

8.Common Mistakes While Identifying Opportunities

☐ Choosing problems without users

☐ Selecting problems that are too small or too large

☐ Ignoring feasibility

"Good opportunity is meaningful, solvable, and valuable."

9.Key Takeaways

- Opportunity areas come from real problems
- Users are the source of opportunities
- Observation and empathy are key
- Correct opportunity reduces innovation risk
- Designers convert problems into value

Period 4

Interviewing Techniques

1.What is Interviewing in Design Thinking?

Definition:

Interviewing is a structured conversation with users to understand:

- Their problems
- Their needs
- Their behavior
- Their emotions and experiences

Interviewing in Design Thinking is **not like an exam** and **not like a police investigation**.

It is:

- Friendly
- Open
- Curious
- Non-judgmental

Simple Explanation for Students:

"Interviewing means talking to users to learn from their experiences."

2.Why Interviewing is Important in Design Thinking

Designers often **assume** problems.
But assumptions are dangerous.

Interviewing helps to:

- Avoid assumptions
- Discover real problems
- Understand user perspective
- Reduce validation risk

Important line:

"Users are experts of their own problems."

3.When Do We Conduct Interviews?

- Before designing solutions
- During problem validation
- While testing ideas or prototypes

"Interviewing is not a one-time activity; it happens throughout the design process."

4.Types of Interview Questions

1. Closed-Ended Questions ☐

Definition:
Questions with **Yes / No / One-word answers**

Examples:

- Do you like this app?
- Is this system useful?

Problem with Closed Questions:

- Very little information
- No deep insights

"Closed questions stop the conversation."

2. Open-Ended Questions ☐

Definition:
Questions that encourage users to **explain their experience**

Examples:

- Can you describe your experience using this app?
- What challenges do you face while using this service?

Why Open Questions are better:

- Users speak more
- Hidden problems come out
- Emotional insights are revealed

"Always ask 'HOW' and 'WHY' questions."

5.Types of Interviews

1. Structured Interviews

- Fixed set of questions
- Same questions for all users
- Easy to compare answers

Limitation:

- Less flexibility

2. Semi-Structured Interviews (Recommended)

- Prepared questions
- Follow-up questions allowed
- Natural conversation

"This is the best type for Design Thinking."

3. Unstructured Interviews

- No fixed questions
- Free conversation
- Deep insights but difficult to analyse

6.Interviewing Process – Step by Step

Step 1: Prepare for the Interview

- Decide who to interview
- Define purpose
- Prepare open-ended questions

Example:
Interview college students to understand online learning problems.

Step 2: Conduct the Interview

- Make the user comfortable
- Be polite and friendly
- Listen more, talk less
- Do not interrupt

"The interviewer should listen, not lecture."

Step 3: Observe Non-Verbal Cues

- Facial expressions
- Tone of voice
- Body language

"Sometimes users don't say the problem, but show it."

Step 4: Record Insights

- Take notes
- Write key pain points
- Capture quotes

✓ Do not write everything—write insights.

7.Do's and Don'ts of Interviewing

☐ **Do's**

- Ask open questions
- Be neutral
- Encourage storytelling
- Be patient

☐ Don’ts

- Do not judge
- Do not lead the user
- Do not suggest solutions
- Do not rush

8. Real-Life Example

Example: Interviewing for College Canteen Improvement

Bad Question ☐:
Do you like the food in the canteen?

Good Question ☐:
Can you describe your experience eating at the canteen?

Insights Discovered:

- Long waiting time
- Limited healthy options
- Seating issues

“Good questions reveal real problems.”

9.Classroom Activity

Activity: Role-Play Interview

Steps:

1. One student = Interviewer
2. One student = User
3. One student = Observer

Topic:

Problems faced during online classes.

Observer notes:

- Good questions
- Mistakes

Purpose:

- Practice real interviewing
- Build confidence

10.Common Interviewing Mistakes

☐ Asking leading questions
☐ Talking too much
☐ Ignoring emotions
☐ Jumping to solutions

> *"Interviewing is about understanding, not selling ideas."*

11. Key Takeaways

- Interviews uncover real problems
- Open-ended questions are powerful
- Listening is more important than speaking
- Interviews reduce innovation risk
- Good interviews lead to good design

Period 5

Empathy-Building Techniques

1.What is Empathy?

Definition:

Empathy is the ability to understand and share the feelings, thoughts, and experiences of another person.

"Empathy means putting yourself in the user's place."

Empathy vs Sympathy

Sympathy	Empathy
Feeling sorry	Understanding deeply
Emotional	Experiential

*Empathy leads to **better design decisions**.*

2.Why Empathy is Critical in Design Thinking

Users may:

- Hide frustration
- Accept inconvenience
- Blame themselves

Empathy helps designers:

- See real struggles
- Understand emotional pain
- Design meaningful solutions

Key line:

“Without empathy, design becomes guesswork.”

3.Empathy-Building Techniques

1. Observation

What it means:

- Watching users in real situations
- Not interfering

Example:
Observe students during exam registration.

Why important?

- People behave differently than they say

2. Empathy Map

Empathy map includes:

- Says
- Thinks
- Does
- Feels

This helps organize user insights visually.

3. User Shadowing

Meaning:

- Follow users during an activity
- Experience their routine

Example:
Accompany a patient in a hospital.

4. Experience Prototyping

Meaning:

- Designer experiences the problem personally

Example:
Using public transport daily to design a travel app.

4. Real-Life Example

Example: Designing for Visually Impaired Users

Empathy actions:

- Designers close eyes while using apps
- Identify difficulties
- Simplify navigation
- Add voice support

"Empathy transforms understanding into action."

5. Classroom Activity – Empathy Exercise

Activity: "Feel the User"

Task:

- Students imagine being:
  - ✓ Elderly user
  - ✓ Rural user
  - ✓ Disabled user

Question:

- What difficulties will they face?

Purpose:

- Build emotional understanding
- Improve inclusive thinking

6. Common Mistakes Without Empathy

☐ Designing for oneself
☐ Ignoring emotional needs
☐ Over-engineering

Empathy avoids these mistakes.

7.Key Takeaways

- Empathy is the foundation of HCD
- Observation reveals hidden problems
- Empathy maps organize insights
- Better empathy = better solutions

Period 6

Mitigate Validation Risk With FIR (Forge Innovation Rubric)

1.What is Validation Risk?

Validation risk is the **possibility that the product or solution will fail** because:

- The **problem is not important** to users
- The **solution is not useful or usable**

- Users **do not adopt it**

"Imagine designing a new type of pen that writes upside down. Technically amazing, but no one needs it. This is validation risk."

2.Types of Validation Risk

Type	Explanation	Example
Problem Risk	Are we solving a real problem?	Do students really need an app to track lecture notes, or is this redundant?
Solution Risk	Does the solution solve the problem effectively?	Does the note-taking app really make note-taking easier?
Market/User Adoption Risk	Will users accept or use the solution?	Will students actually download and use the app regularly?

"Even if you build the perfect product, ignoring users can lead to failure."

3.What is FIR – Forge Innovation Rubric?

Definition:

FIR is a **structured framework** to **evaluate ideas and prototypes early**.
It helps teams **mitigate validation risks** by checking assumptions systematically.

Key Idea:

FIR is like a **checklist for innovation**, ensuring the idea is:

- Relevant
- Valuable
- Feasible
- Tested with users

4. FIR Evaluation Parameters

Parameter	What it Means	Classroom Example
Problem Clarity	Is the problem clearly defined?	Students can state: "Students struggle to find empty classrooms."
User Relevance	Does the problem matter to users?	Ask students: "Do users feel this is an important issue?"
Value Proposition	Does the solution provide clear benefit?	App notifies available classrooms → saves time
Feasibility	Can the solution be realistically implemented?	Can the app be developed with available resources?
Validation Evidence	Is there proof that users want this solution?	Conduct survey/interviews before building the app

5. How FIR Helps Mitigate Validation Risk

1. **Structured Assessment:** FIR gives clear parameters to check ideas.
2. **Early Feedback:** Identify problem or solution gaps before full development.
3. **Prioritization:** Helps decide which ideas are worth investing in.
4. **Reduced Failure:** Only solutions that pass FIR move to prototyping.

Analogy:

*"Think of FIR as a **safety net for innovation**. You check your steps before taking a leap."*

6.Real-Life Example

Example: Swiggy Food Delivery App

- Problem risk: People want food delivered → real problem ☐

- Solution risk: Easy app with ordering and tracking → solves problem ☐
- Market risk: Users adopt → high demand ☐
- FIR applied informally: Teams assess if the idea is clear, relevant, feasible, and desired → reduces risk

7.Classroom Activity

Activity: Apply FIR to a Student Idea

Steps:

1. Students form small teams
2. Identify one problem they want to solve
3. Use FIR parameters to evaluate the idea:
    - ✓ Problem clarity
    - ✓ User relevance
    - ✓ Value proposition
    - ✓ Feasibility
    - ✓ Validation evidence
4. Discuss which ideas are **strong and which need revision**

Purpose:

- Hands-on experience with structured validation
- Understand why some ideas fail early

8.Common Mistakes Without FIR

☐ Jumping straight to building the solution
☐ Ignoring problem importance
☐ Assuming users will adopt without evidence

"FIR prevents teams from wasting time, money, and effort on unvalidated ideas."

9.Key Takeaways

- Validation risk is the risk of **solution failure due to unvalidated assumptions**
- FIR is a **framework to check ideas systematically**
- Parameters: **Problem clarity, User relevance, Value proposition, Feasibility, Validation evidence**
- Using FIR early **reduces failure risk and improves success chances**

Period 7

FIR – Components and Application

1.What is FIR?

- FIR is a **tool/framework** to systematically assess innovation ideas.
- It **mitigates risks** early in the innovation process.
- Ensures that the solution is **relevant, feasible, valuable, and validated** before investing in full-scale development.

*"Think of FIR as a **checklist for innovation success**."*

2.Components of FIR

FIR evaluates ideas based on **five key components**:

Component	Explanation	Example (Student Context)
1. Problem Clarity	Is the problem clearly defined and understandable?	Problem: Students waste time finding empty classrooms
2. User Relevance	Does the problem matter to the target user?	Users (students) say this is a major pain point
3. Value	Does the solution provide	App notifies available

Component	Explanation	Example (Student Context)
Proposition	real benefit or value?	classrooms → saves time
4. Feasibility	Can the solution be implemented with available resources?	Can the app be developed within college resources and skillset?
5. Validation Evidence	Is there proof that users want and will adopt the solution?	Surveys, interviews, or testing show students would use the app

Explanation Tips:

- Explain **Problem Clarity**: "Without a clear problem, solutions will be wasted."
- Explain **User Relevance**: "Even a great solution is useless if users don't care."
- Explain **Value Proposition**: "Users must feel the benefit clearly."
- Explain **Feasibility**: "Technical or resource constraints must be considered."
- Explain **Validation Evidence**: "Proof from users reduces risk."

3.How to Apply FIR

1. **Identify the Problem**
   - ✓ Clearly define the problem using observation and interviews.
2. **Define Target Users**
   - ✓ Who is affected by the problem?
   - ✓ Understand their needs and priorities.
3. **Generate Solution Ideas**
   - ✓ Brainstorm multiple alternatives
   - ✓ Shortlist ideas that solve the problem
4. **Evaluate Using FIR Components**
   - ✓ Check each idea for:
     - Problem clarity
     - User relevance
     - Value proposition

- Feasibility
- Validation evidence

5. **Decide on Best Idea**
   - ✓ Choose ideas that score high across FIR
   - ✓ Revise ideas that score low

4. Real-Life Example – Applying FIR

Scenario: College Classroom Finder App

FIR Component	Application
Problem Clarity	Students often cannot find empty classrooms for study groups
User Relevance	Most students report wasting 10–15 mins per day searching for rooms
Value Proposition	Saves time, reduces frustration, improves productivity
Feasibility	College Wi-Fi + simple app + admin support makes implementation possible
Validation Evidence	Survey shows 80% of students would use it

5.Classroom Activity

Activity: FIR Evaluation Exercise

1. Form small groups (3–4 students)
2. Each group chooses one problem they want to solve
3. Apply **all five FIR components** to evaluate their solution idea
4. Discuss results with the class

Purpose:

- Students learn **practical application** of FIR

- They understand how to **filter strong ideas from weak ones**

6.Common Mistakes When Using FIR

☐ Ignoring one or more components
☐ Skipping user validation
☐ Overestimating feasibility without checking resources

"FIR only works if all components are considered honestly."

7.Key Takeaways

- FIR is a **structured rubric for validating innovation ideas**
- Components: **Problem Clarity, User Relevance, Value Proposition, Feasibility, Validation Evidence**
- Applying FIR helps **mitigate validation risk**
- Strong FIR application leads to **better prototypes and higher chance of success.**

Period 8

Case Studies – Part I

1. Why Case Studies are Important

- Helps students **understand practical applications**
- Shows how **concepts like HCD, FIR, and prototyping** work in real life
- Encourages **critical thinking**
- Provides inspiration for **student projects and ideas**

"Learning from success and failure in real cases reduces risk in your own projects."

2. Case Study 1: UPI – Unified Payments Interface (India)

Step 1: The Problem

- Users faced difficulties:
  - ✓ Long queues at banks
  - ✓ Multiple bank accounts for transfers
  - ✓ Slow transactions

Step 2: Human-Centered Approach

- Designed **simple app interfaces** for transactions
- Focused on **ease of use and accessibility**
- Iteratively tested with real users

Step 3: Innovation Process Applied

- Problem identification: Slow banking processes
- Ideation: Instant digital payments across banks
- Prototyping: Early app versions with basic functionality
- Testing & iteration: Added UPI ID, QR codes, and multiple bank links

Step 4: Validation & FIR Application

- Problem Clarity: Clear – reduce time for payments ☐
- User Relevance: Very high – millions need it ☐
- Value Proposition: Simple, instant payments ☐
- Feasibility: Digital platforms + banks cooperation ☐
- Validation Evidence: Pilot tests with users, adoption rates ☐

Step 5: Result

- Widely adopted across India
- Millions of daily transactions
- Simplified digital payments

"UPI is a great example of applying HCD, FIR evaluation, and iterative innovation."

3. Case Study 2: OLA/UBER – Ride-Hailing Services

Step 1: The Problem

- Urban commuters faced:
  - ✓ Unreliable taxis
  - ✓ Long wait times
  - ✓ Unsafe rides

Step 2: Human-Centered Approach

- Simple app with GPS and live tracking
- Focused on **user convenience and safety**

Step 3: Innovation Process

- Problem identification: Need for safe, reliable transport
- Ideation: Book rides via app, driver tracking
- Prototyping: Initial app with few features
- Testing & iteration: Added features like fare estimate, ratings, cashless payment

Step 4: Validation & FIR Application

- Problem Clarity: Clear – commuting pain points ☐
- User Relevance: High – daily commuters ☐

- Value Proposition: Reliable, convenient, safe travel ☐
- Feasibility: Mobile apps + GPS + driver network ☐
- Validation Evidence: Initial pilots in few cities, user adoption feedback ☐

Step 5: Result

- Transformed urban commuting
- Millions of users globally
- Continuous improvement and feature upgrades

"Both UPI and OLA/UBER show the importance of understanding users, iterating solutions, and validating ideas."

4. Classroom Activity

Activity: Analyze a Case Study

Steps:

1. Divide students into small groups
2. Provide one product/service (e.g., Swiggy, Flipkart, Google Maps)
3. Ask them to identify:
    - ✓ Problem solved
    - ✓ How HCD was applied
    - ✓ FIR evaluation (if applicable)
    - ✓ Result or impact

Purpose:

- Helps students **connect theory to practice**
- Encourages **critical thinking**
- Sparks **project ideas**

5. Common Observations from Case Studies

Successful innovations **start with real problems**

- **Empathy and user understanding** are key
- Iteration is crucial for improvement
- FIR or structured evaluation **reduces risk**
- Early prototyping and feedback help scale faster

5. Key Takeaways

- Case studies provide **learning from real examples**
- HCD, Innovation Process, and FIR are **interconnected**
- Observing **success and failure** guides better design decisions
- Students can apply these principles in **projects and competitions**

Period 9

Case Studies – Part II

'Case Studies – Learning from successes and failures in innovation'

1.Why Continue Case Studies?

- Reinforces **practical understanding**
- Shows **different types of innovation**
- Helps students **think critically about user needs**
- Encourages **application in student projects**

"By seeing patterns across multiple cases, students understand what makes innovation successful."

2. Case Study 3: Swiggy – Online Food Delivery

Step 1: Problem

- Users had difficulties:
  - ✓ Ordering food over phone
  - ✓ No tracking of orders
  - ✓ Unreliable delivery times

Step 2: Human-Centered Design

- Developed **simple mobile app interface**
- Real-time order tracking
- Focused on **ease of use and reliability**

Step 3: Innovation Process

- Ideation: Multiple ordering methods
- Prototyping: Basic app version for limited restaurants
- Testing & Iteration: Added features – live tracking, payment options, ratings

Step 4: FIR Application

- Problem Clarity: Clear – user wants faster, reliable food delivery ☐
- User Relevance: High – busy urban population ☐
- Value Proposition: Convenience, speed, choice ☐
- Feasibility: App development + delivery network ☐
- Validation Evidence: Early beta testing and user feedback ☐

Step 5: Result

- Millions of users adopted the app
- Became a leading online food delivery platform

3.Case Study 4: Google Maps – Navigation and Location Services

Step 1: Problem

- Difficulty in finding routes and locations
- Paper maps were slow and inaccurate

Step 2: Human-Centered Approach

- Simple, interactive map interface
- Voice-guided navigation
- Real-time traffic updates

Step 3: Innovation Process

- Ideation: Combine GPS, satellite imagery, and real-time data
- Prototyping: Early map app with basic features
- Testing: Feedback from users for accuracy and usability
- Iteration: Added route suggestions, offline maps, traffic alerts

Step 4: FIR Application

- Problem Clarity: Clear – people need accurate navigation ☐
- User Relevance: High – every driver or traveler ☐
- Value Proposition: Save time, reduce confusion ☐
- Feasibility: Technology + data integration ☐
- Validation Evidence: Beta users and continuous feedback ☐

Step 5: Result

- Became **worldwide navigation standard**
- Continuous updates improve user experience

4.Classroom Activity

Activity: Compare Two Case Studies

Steps:

1. Divide students into two groups
2. Assign one group Swiggy, one group Google Maps
3. Ask students to analyze:
    - ✓ Problem identification
    - ✓ HCD principles applied
    - ✓ FIR evaluation
    - ✓ Innovation process
4. Each group presents **key learnings** to the class

Purpose:

- Reinforces understanding
- Encourages critical thinking
- Helps students **apply concepts to new ideas**

5.Common Observations Across Case Studies

- Successful innovations **start with real user problems**
- Human-Centered Design is **central to adoption**
- Iterative prototyping and feedback are crucial
- FIR helps **mitigate risk and validate ideas early**
- Multidisciplinary collaboration often drives success

6.Key Takeaways

- Case studies provide **practical insight**
- Concepts like **HCD, innovation process, prototyping, and FIR** are **interconnected**
- Understanding **user needs** is always the first step

- Early testing and validation reduce failure risk
- Students can **apply these frameworks to their projects**

UNIT II

ENDUSER-CENTRIC INNOVATION

Period 1

Importance of Customer-Centric Innovation

1.What is Customer-Centric Innovation?

Definition:

Customer-Centric Innovation is the process of **developing products, services, or solutions by prioritizing the needs, preferences, and experiences of the customer**.

Key Point:

- Customers are **at the center** of every design and business decision.
- Success depends on **solving real problems that customers care about**.

2.Why is Customer-Centric Innovation Important?

1. **Reduces Risk of Failure**
   - ✓ Designing without understanding customers often leads to rejection.
2. **Creates Real Value**
   - ✓ Focused on solving **problems that matter**.

3. **Enhances Adoption & Loyalty**
    - ✓ Customers are more likely to use and recommend products they feel are designed for them.
4. **Encourages Continuous Feedback**
    - ✓ Customers guide **improvements and iterations**.

3. Difference Between Product-Centric and Customer-Centric Approaches

Product-Centric Innovation	Customer-Centric Innovation
Focus on technology or features	Focus on solving customer problems
Assumes what the customer wants	Validates what the customer truly needs
Risk of rejection is high	Higher adoption and satisfaction
Improvement may be internal-driven	Improvement guided by customer feedback

Example:

"A company may build a phone with many features (product-centric), but users may only want a long battery life and simple interface (customer-centric)."

4.Key Principles of Customer-Centric Innovation

1. **Empathize with Customers** – Understand their pain points, behavior, and desires.
2. **Problem Validation** – Ensure the problem is real and significant.
3. **Iterate Based on Feedback** – Adjust solution based on what customers actually want.
4. **Measure Impact** – Check whether the solution **creates value** for the customer.

5.Real-Life Example

Example: Flipkart Cash-on-Delivery (India)

Problem:

- Customers were hesitant to pay online due to trust issues.

Customer-Centric Solution:

- Introduced **Cash-on-Delivery**
- Focused on customer preference and trust
- Tested and refined based on feedback

Result:

- Increased adoption
- Improved customer satisfaction and sales

"This shows how listening to customers and designing solutions around their needs leads to success."

6.Classroom Activity

Activity: Identify Customer-Centric Opportunities

Steps:

1. Ask students to think of a daily problem they face as a customer (e.g., ordering food, commuting, studying).
2. Identify what **solution could help**.
3. Discuss: Is the solution focused on **real customer needs or assumptions?**

Purpose:

- Reinforces importance of **customer focus**

- Encourages **empathy and critical thinking**

7.Common Mistakes Without Customer-Centric Innovation

☐ Designing based on assumptions

☐ Focusing only on technology or features

☐ Ignoring user feedback

"Customer-centric innovation prevents building products no one wants."

8.Key Takeaways

- Customer-Centric Innovation **starts with the customer, not the product**
- Empathy, problem validation, and feedback are critical
- It **reduces failure risk** and **increases adoption**
- Real-life examples (Flipkart COD, UPI, Ola) show its impact

---

Period 2

Problem Validation

1.*What is Problem Validation?*

Definition:

Problem Validation is the process of **confirming that a perceived problem is real, important to users, and occurs frequently enough to warrant a solution.**

Key Points:

- Avoids building solutions for problems that **don't exist**
- Helps **prioritize ideas** based on impact

- Reduces **wasted time, effort, and resources**

2.Why Problem Validation is Important

1. **Avoids Assumptions:**
   - ✓ Designers often assume they know the problem; validation proves it.
2. **Prioritizes Effort:**
   - ✓ Focus resources on solving **high-impact problems**.
3. **Reduces Risk:**
   - ✓ Early validation reduces **product failure**.
4. **Guides Solution Design:**
   - ✓ Understanding the problem clearly helps design a **better solution**.

3.Steps to Validate a Problem

1. **Identify the Problem**
   - ✓ Observe users
   - ✓ Conduct interviews
   - ✓ Analyze complaints or feedback
2. **Measure Problem Significance**
   - ✓ How severe is the problem?
   - ✓ How much pain does it cause the user?
3. **Measure Problem Incidence**
   - ✓ How often does it happen?
   - ✓ Is it a widespread issue or rare?
4. **Gather Evidence from Real Users**
   - ✓ Surveys, interviews, and observations
   - ✓ Look for patterns in user responses
5. **Refine or Reject Problem**
   - ✓ If validation fails, **do not proceed**
   - ✓ Only move forward with **validated problems**

4.Example – Student Scenario

Problem: Students waste time finding empty classrooms.

Validation Process:

- **Observation:** Check how many students search for rooms daily
- **Interviews:** Ask 20–30 students about their experience
- **Survey:** Measure the frequency and impact of the problem

Outcome:

- If 80% report frustration, problem is validated
- If only 5% face the issue, reconsider solving this problem

"*Problem validation ensures we solve* ***problems that really matter****.*"

5.Classroom Activity

Activity: Validate a Problem

Steps:

1. Ask students to identify **one problem** they face in daily life (school, commuting, apps)
2. Discuss:
    - ✓ Who experiences this problem?
    - ✓ How significant is it?
    - ✓ How often does it occur?
3. Decide whether the problem is **worth solving**

Purpose:

- Students learn to **validate before designing**
- Encourages **observation and critical thinking**

6.Common Mistakes in Problem Validation

☐ Assuming a problem exists without evidence
☐ Only talking to friends or people similar to you
☐ Ignoring frequency or impact of the problem

*"Validation is not about **your opinion**, it's about **real user evidence**."*

7.Key Takeaways

- Problem Validation ensures the problem is **real, significant, and widespread**
- Focus on **user evidence** to guide decision-making
- Skipping validation increases risk of **building unwanted solutions**
- Critical for **customer-centric innovation**

Period 3

Customer Discovery

"Customer Discovery = Learning who your customers are, their needs, and behaviors."

1.What is Customer Discovery?

Definition:

Customer Discovery is the process of **actively engaging with potential users or customers to understand their problems, needs, preferences, and behaviors**.

Key Points:

- Focus on **learning, not selling**
- Helps in **building empathy** with users

- Identifies **hidden insights** that might not appear in surveys

2.Why Customer Discovery is Important

1. **Identifies Target Users:**
   - ✓ Who are the real people affected by the problem?
2. **Uncovers Hidden Needs:**
   - ✓ Sometimes users **can't articulate their problems clearly**
3. **Guides Solution Design:**
   - ✓ Helps design **solutions aligned with real behaviors and preferences**
4. **Reduces Risk of Failure:**
   - ✓ Avoids building products for the wrong audience

3.Steps in Customer Discovery

1. **Define Assumptions:**
   - ✓ List what you **assume about your customers and their problems**
2. **Identify Target Customers:**
   - ✓ Who is most affected by the problem?
   - ✓ Consider demographics, occupation, lifestyle, etc.
3. **Engage with Customers:**
   - ✓ Interviews, surveys, observations, or shadowing
   - ✓ Focus on **listening, not convincing**
4. **Capture Insights:**
   - ✓ Document pain points, behavior patterns, and emotional triggers
5. **Refine Understanding:**
   - ✓ Adjust assumptions based on what you learn
   - ✓ Repeat until you clearly **understand the customer**

4.Example - Parking Scenario

Problem: People waste time finding parking spaces

Customer Discovery Process:

• **Define assumptions:**
"Drivers spend 10 minutes searching for parking."

• **Identify target users:**
Students and staff who drive to campus.

• **Engage:**
Ask:

- How often do you struggle to find parking?
- How long does it take?

• **Capture insights:**
Most drivers face this problem daily, especially in the morning.

• **Refine:**
Some use public transport, so solutions should focus on regular drivers.

Outcome:
Better understanding of users and their parking problems.

7.Classroom Activity

Activity: Conduct Mini Customer Discovery

Steps:

1. Form pairs or small groups
2. Each group selects a **problem in student life**

3. Conduct **mini-interviews** with 3–5 classmates:
    - ✓ Ask open-ended questions
    - ✓ Focus on pain points and experiences
4. Document insights and discuss in class

Purpose:

- Hands-on experience in **learning from users**
- Builds **empathy and listening skills**

8. Common Mistakes in Customer Discovery

☐ Talking to people outside the target audience
☐ Asking leading questions
☐ Assuming insights without verification

*"The goal is to **observe and listen**, not to confirm your assumptions."*

9.Key Takeaways

- Customer Discovery is **about understanding the user deeply**
- Focus on **learning, empathy, and observation**
- Validates **target user, needs, and behaviors**
- Critical for **customer-centric innovation and effective solutions**

Period 4

Understanding Problem Significance and Problem Incidence

"Problem Significance = How important the problem is to users"

"Problem Incidence = How often the problem occurs or affects users"

1.What is Problem Significance?

Definition:

Problem Significance is the **importance or impact of a problem on the user's life or experience**.

Key Points:

- Significant problems cause **real pain, frustration, or inconvenience**
- High-significance problems are worth solving because solutions create **real value for users**

Example:

- **Low significance:** Choosing a pen color
- **High significance:** Students wasting hours finding study rooms

"The bigger the pain or consequence, the more significant the problem is."

2.What is Problem Incidence?

Definition:

Problem Incidence is **how frequently the problem occurs or how many users are affected**.

Key Points:

- High incidence means the problem **affects many users or occurs frequently**
- Helps **prioritize solutions** based on scale

Example:

• **Low incidence:** A student's laptop charger breaks once in a year.

• **High incidence:** Most students face slow campus Wi-Fi every day.

"Even a significant problem may not be worth solving if it happens rarely."

3.Why Problem Significance and Incidence Matter

1. **Prioritize Effort and Resources:**
   - ✓ Focus on problems that **matter most to many users**
2. **Maximize Impact:**
   - ✓ Solutions targeting high-significance and high-incidence problems **deliver more value**
3. **Reduce Failure Risk:**
   - ✓ Avoid investing in problems that **users don't care about**

4.Steps to Evaluate Problem Significance and Incidence

1. **Gather Data from Users**
   - ✓ Interviews, surveys, observations
2. **Score the Problem**
   - ✓ Significance: Rate severity (1–5)
   - ✓ Incidence: Rate frequency (1–5)
3. **Prioritize Problems**
   - ✓ High significance + High incidence = **top priority**
   - ✓ Low significance or low incidence = **consider deprioritizing**

4. Example

Problem: People wait a long time in hospital queues

Evaluation:

• **Significance:** High (wastes time, causes stress)

• **Incidence:** High (many patients face it daily)

Decision:

• High priority for solution development

Contrast Example

Problem: Choosing background music in a waiting room

Evaluation:

• **Significance:** Low

• **Incidence:** Low

Decision:

• Not worth designing a product for

5.Classroom Activity

Activity: Score Problems for Significance and Incidence

Steps:

1. Form small groups
2. Each group lists **3–5 common student problems**
3. Assign scores for:
    - ✓ Problem significance (1–5)
    - ✓ Problem incidence (1–5)
4. Identify **high-priority problems** for innovation

Purpose:

- Teaches **prioritization based on impact and frequency**
- Prepares students for **solution design based on real user needs**

6.Common Mistakes

☐ Ignoring how often the problem occurs
☐ Focusing only on personal opinion, not data
☐ Solving low-impact problems first

"A problem may be painful for one person, but if it doesn't affect many users, it's not high priority."

7.Key Takeaways

- **Problem Significance:** Measures **importance of the problem**
- **Problem Incidence:** Measures **frequency and reach of the problem**
- Together, they **help prioritize which problems to solve**
- High significance + high incidence = **best focus for innovation**

Period 5

Understanding Problem Incidence

"*Problem Incidence = Frequency and reach of the problem among users.*"

1.What is Problem Incidence?

Definition:

Problem Incidence is the **measure of how frequently a problem occurs or how many users experience it.**

Key Points:

- Helps **quantify the scale of the problem**
- High-incidence problems **affect many users or occur frequently**, making them ideal for innovation

2.Why Problem Incidence is Important

1. **Prioritization:**
   - ✓ Focus on problems that **impact more users or occur more often**
2. **Resource Optimization:**
   - ✓ Solutions for high-incidence problems **benefit more people**
3. **Decision Making:**
   - ✓ Helps innovators **choose which problems to solve first**
4. **Risk Reduction:**
   - ✓ Low-incidence problems may **not justify the effort**, reducing unnecessary investment

3.How to Measure Problem Incidence

1. **Observation:**
   - ✓ Watch how often users encounter the problem
2. **Surveys and Interviews:**
   - ✓ Ask users about **frequency of occurrence**
3. **Data Analysis:**
   - ✓ Check usage patterns, complaints, or behavioral data

4. **Scoring Incidence:**
    - ✓ Example scale:
        - 1 = Rare (affects few users or occurs occasionally)
        - 3 = Moderate (affects several users occasionally)
        - 5 = High (affects most users frequently)

4.Example – General Scenario

Problem: Traffic congestion during office hours

Assessment:

- **Frequency:** Happens every working day
- **Affected Users:** Large number of daily commuters

Incidence Score: High (5/5)

Contrast Example

Problem: Misplacing a water bottle once in a while

Assessment:

- **Frequency:** Rare
- **Affected Users:** Very few people

Incidence Score: Low (1/5)

"Even if a problem is significant, if it occurs rarely, it may not be worth prioritizing."

5.Classroom Activity

Activity: Evaluate Problem Incidence

Steps:

1. Students form small groups

2. List **3–5 problems** they face daily as students
3. Assign **incidence scores (1–5)** for each problem based on frequency and affected users
4. Identify **high-incidence problems** for potential solutions

Purpose:

- Teaches students **how to quantify problem impact**
- Prepares them for **data-driven decision making in innovation**

6.Common Mistakes

☐ Assuming problems are widespread without evidence
☐ Ignoring frequency in favor of severity only
☐ Overestimating incidence based on personal experience

"*Validation requires **objective evidence**, not assumptions or anecdotes.*"

7.Key Takeaways

- **Problem Incidence measures frequency and reach** of a problem
- **High-incidence problems** are more impactful to solve
- **Combine incidence with significance** to prioritize solutions
- Helps **allocate resources effectively** in innovation

Period 6

Customer Validation

"Customer Validation = Confirming that target customers truly have the problem and are likely to adopt a solution."

1. What is Customer Validation?

Definition:

Customer Validation is the process of **testing and confirming assumptions about the target users and their needs** to ensure that the problem and solution are aligned with real customer behavior.

Key Points:

- Prevents designing for the **wrong audience**
- Confirms **problem importance and relevance**
- Helps refine **user persona and solution design**

2.Why Customer Validation is Important

1. **Reduces Risk:**
   - ✓ Ensures you are building for customers who **actually need it**
2. **Confirms Product-Market Fit:**
   - ✓ Helps determine if the solution **solves the right problem for the right audience**
3. **Refines Understanding:**
   - ✓ Reveals gaps or misconceptions about user needs
4. **Guides Solution Development:**
   - ✓ Provides insights for **prototype design and features**

3.Steps in Customer Validation

1. **Define Hypotheses:**
   - ✓ List assumptions about users, their problems, and your proposed solution
2. **Develop a Validation Plan:**

- ✓ Decide how to test hypotheses (interviews, surveys, field observation)

3. **Engage with Target Customers:**
   - ✓ Ask open-ended questions
   - ✓ Observe behavior rather than just opinions
4. **Analyze Feedback:**
   - ✓ Identify patterns and confirm or reject assumptions
5. **Iterate:**
   - ✓ Refine user persona, problem understanding, or solution based on findings

4.Example

Problem: People often forget to drink enough water during the day

Customer Validation Process:

• **Hypothesis:**
Most people forget to drink water regularly

• **Validation Method:**
Interview 20 people and observe their daily habits

• **Findings:**
o 15 out of 20 people said they forget to drink water
o Many only drink water when they feel thirsty

• **Conclusion:**
The hypothesis is validated — forgetting to drink water is a real and common problem

Simple Explanation for Students

In this example:

- We **assumed** a problem (people forget to drink water)
- We **talked to real users**

- We **collected proof**
- We **confirmed** the problem exists

So, **customer validation helps decide whether a problem is real and worth solving**.

5.Classroom Activity

Activity: Validate a Problem with Peers

Steps:

1. Form pairs or small groups
2. Each group selects **a validated problem** from previous exercises
3. Conduct mini-validation by interviewing 3–5 classmates:
    - ✓ Ask about frequency, severity, and current solutions
4. Document insights and discuss if the problem is **confirmed**

Purpose:

- Provides **hands-on experience in validating assumptions**
- Reinforces importance of **real user evidence**

6.Common Mistakes

☐ Assuming a problem is real without customer feedback
☐ Talking only to friends or biased users
☐ Focusing on what **you think** users want rather than evidence

*"Validation requires **objective user feedback**, not assumptions."*

7.Key Takeaways

- Customer Validation ensures **the problem is real and relevant**
- Confirms **target user, needs, and solution fit**

- Critical for **reducing failure risk** and **building user-centric solutions**
- Forms the foundation for **designing user personas and stories**

Period 7

Target User & User Persona

"Target User = Specific group of people for whom the solution is designed"

"User Persona = A detailed fictional representation of the target user"

1.What is a Target User?

Definition:

Target Users are the specific group of people most affected by the problem and most likely to use your solution.

Key Points:

- Focus on who experiences the problem most
- Consider demographics, behavior, needs, and context
- Helps prioritize design decisions

Example:

Problem: Difficulty finding available parking on campus

Target Users: Students and staff who drive to campus regularly

Explanation:
These users are most affected by the problem and most likely to use a solution like a parking availability app or real-time parking updates

2.What is a User Persona?

Definition:

A User Persona is a fictional, detailed representation of a typical target user, based on research and validation.

Key Components of a User Persona:

Component	Description	Example
Name	Fictional name for identification	"Rohan, 2nd Year Engineering Student"
Demographics	Age, gender, occupation, location	20 years old, male, Bangalore
Goals	What the user wants to achieve	Find available classrooms quickly
Pain Points	Challenges the user faces	Wastes 20–30 mins daily searching for study rooms
Behaviors	How the user interacts with the environment	Uses mobile apps, checks notice boards, asks peers
Preferences	User likes/dislikes	Prefers digital notifications over word-of-mouth

"Personas make the target user **real and relatable** for design decisions."

3.Why Target Users and Personas are Important

1. **Guides Solution Design:**
   - ✓ Helps focus on **real needs, not assumptions**
2. **Improves Communication:**
   - ✓ Personas allow the **team to discuss user needs consistently**
3. **Supports Testing:**
   - ✓ Makes **prototype testing** more realistic
4. **Enhances Empathy:**
   - ✓ Team members understand **users' experiences and motivations**

4. Example

Problem: Difficulty tracking personal expenses

- This is the issue the solution will solve.

Target User: Young adults managing their first budget

- This shows **who is most affected** by the problem.
- Target users are the people who will **benefit most** from the solution.

User Persona: Sara, age 24

- **Name and age** make the persona feel real and relatable.
- **Goal:** "Keep track of daily expenses and save money" → tells what the user wants to achieve.
- **Pain Points:** "Often forgets where money was spent, overspends, feels stressed" → tells what frustrates the user.
- **Behavior:** "Uses notes or spreadsheets inconsistently, rarely checks banking apps" → shows how the user currently deals with the problem.
- **Preference:** "Prefers a simple mobile app that automatically categorizes expenses and sends reminders" → shows what kind of solution the user would like.

Why this matters

- **Sara represents many young adults** with the same challenges.
- By designing the solution with Sara in mind, the team knows exactly **what features are important**, such as:
  - ✓ Automatic expense tracking
  - ✓ Simple notifications
  - ✓ Easy-to-understand dashboard

- Without a persona, the team might **guess what users need**, which often fails.

5.Classroom Activity

Activity: Create a User Persona

Steps:

1. Form small groups
2. Choose a **validated problem** from previous exercises
3. Identify the **target user group**
4. Create a **detailed persona** including:
    - ✓ Name, demographics, goals, pain points, behaviors, preferences
5. Present persona to the class and discuss design implications

Purpose:

- Students learn to **think from the user's perspective**
- Prepares them for **solution design and prototyping**

6. Common Mistakes

☐ Making personas based on assumptions
☐ Ignoring differences among users
☐ Creating vague or generic personas

"Personas must be data-driven and specific, reflecting real user behavior."

7.Key Takeaways

- **Target Users:** Specific group affected by the problem
- **User Persona:** Fictional representation of the target user, based on research

- Personas guide **design decisions, empathy, and solution testing**
- Helps in creating **customer-centric and effective solutions**

Period 8

User Stories

"*User Story = A simple description of what a user wants to achieve and why"*

1. What is a User Story?

Definition:

A User Story is a short, simple statement describing a feature or requirement from the perspective of the user, focusing on what they want and why it matters.

Structure of a User Story:

As a [type of user], I want [goal] so that [benefit].

Example:

• As a commuter, I want to see real-time bus arrival updates so that I can plan my travel and avoid waiting at the bus stop.

"User Stories are simple sentences that focus on the user's need and the benefit. This helps the team understand what feature to build and why it matters to the user."

2. Why User Stories are Important

1. **Keeps Focus on Users:**
    - ✓ Ensures features solve real user problems

2. **Simple and Clear:**
    - ✓ Easy for the team to understand and implement
3. **Guides Prototyping and Testing:**
    - ✓ Helps define requirements for prototypes
4. **Facilitates Communication:**
    - ✓ Provides a shared understanding among designers, developers, and stakeholders

3.Steps to Create Effective User Stories

1. **Identify the User:**
    - ✓ Use validated target users or personas
2. **Define the Goal:**
    - ✓ What the user wants to achieve
3. **State the Benefit:**
    - ✓ Why this goal matters to the user
4. **Keep it Simple:**
    - ✓ One story should focus on one need at a time
5. **Validate:**
    - ✓ Check with real users if the story truly reflects their needs

4. Example

Problem: Students struggle to find available library computers

- **User Persona:** Akash, 3rd-year student
- **User Story 1:**
  As a student, I want to see which library computers are currently free so that I can start my work without waiting.
- **User Story 2:**
  As a student, I want to reserve a library computer in advance so that I can ensure I have a spot during peak hours.

"Each user story targets a specific need of the user, helping the team design practical solutions that save time and reduce frustration."

5.Classroom Activity

Activity: Create User Stories

Steps:

1. Form small groups
2. Each group chooses a validated problem and a target user/persona
3. Write 3–5 user stories in the format:
    - ✓ As a [user], I want [goal] so that [benefit]
4. Present stories and discuss clarity, focus, and relevance

Purpose:

- Helps students translate user needs into actionable requirements
- Prepares for prototype and solution development

6.Common Mistakes

☐ Writing vague or technical statements instead of user-focused stories
☐ Combining multiple goals in a single story
☐ Ignoring validation with real users

"User stories should always reflect user needs, not what you think is cool to build."

7.Key Takeaways

- User Stories capture user needs in simple, actionable statements
- Structure: As a [user], I want [goal] so that [benefit]
- Guides design, prototyping, and testing
- Ensures solutions are customer-centric and value-driven

Period 9

Customer Development Process – Customer Interviews & Field Visit

"Customer Development = Direct engagement with users to validate problems, needs, and potential solutions."

1.What is the Customer Development Process?

Definition:

The Customer Development Process is a structured approach to interact with target users to discover, validate, and refine problems and solutions.

Key Points:

- Focuses on learning directly from the customer
- Iterative: insights from interviews and observations inform solution design
- Helps ensure customer-centric innovation

2.Steps in Customer Development

1. **Define Objectives:**
   - ✓ Decide what you want to learn (e.g., problem validation, solution feedback)
2. **Identify Target Users:**
   - ✓ Use previously defined target users and personas
3. **Prepare Questions:**
   - ✓ Open-ended, non-leading questions
   - ✓ Avoid yes/no questions
4. **Conduct Customer Interviews:**
   - ✓ Observe behavior, ask questions, and take notes
   - ✓ Focus on understanding pain points, goals, and experiences

5. **Field Visits / Observation:**
    - ✓ Visit places where users interact with products or face problems
    - ✓ Observe behavior in the real environment
6. **Analyze & Document Insights:**
    - ✓ Look for patterns, validate assumptions, and identify opportunities for solutions
7. **Iterate:**
    - ✓ Refine hypotheses, personas, and user stories based on findings

3. Example – Grocery Shopping Scenario

Problem: Shoppers waste time finding products in large supermarkets

Customer Development Process:

- **Objective:** Validate how often and how severely the problem affects shoppers
- **Target Users:** Regular supermarket shoppers
- **Interview Questions:**
    - ✓ How often do you have trouble finding items?
    - ✓ What strategies do you use to locate products?
    - ✓ How much extra time do you spend searching?
- **Field Visit:** Observe shoppers navigating aisles and using store maps
- **Insights:** Most shoppers spend 10–20 minutes extra per visit, especially for unfamiliar sections
- **Next Step:** Design solution (e.g., a mobile app with store maps and real-time product locations)

"This process ensures that solutions are created based on real shopper behavior, making them practical and effective."

4.Classroom Activity

Activity: Conduct Mini Customer Interviews

Steps:

1. Form small groups
2. Select a validated student problem from previous exercises
3. Prepare 3–5 open-ended questions for peers
4. Conduct mini-interviews in the classroom or nearby campus areas
5. Document insights and identify patterns
6. Discuss opportunities for solution design based on findings

Purpose:

- Hands-on experience in customer-centric research
- Prepares students for solution ideation and prototyping
- Reinforces observation, empathy, and interviewing skills

5.Best Practices for Customer Interviews & Field Visits

- Listen more than you speak
- Ask open-ended questions
- Avoid influencing responses
- Take detailed notes or record (with permission)
- Observe context, environment, and behavior
- Respect participants' time and privacy

6.Common Mistakes

 Asking leading or biased questions

☐ Talking only to friends or familiar users

☐ Ignoring behavioral observations

☐ Focusing solely on opinions instead of actions

7. Key Takeaways

- Customer development is hands-on, iterative, and user-focused
- Interviews and field visits validate problems and reveal hidden insights
- Direct engagement improves empathy and understanding
- Provides the foundation for user stories, personas, and solution design

---

UNIT III

Applied Design Thinking Tools

Period 1

Concept of Minimum Usable Prototype (MUP)

"Minimum Usable Prototype (MUP) = A basic version of a product that is usable and testable by users to validate ideas."

1.What is a Minimum Usable Prototype (MUP)?

Definition:

A Minimum Usable Prototype (MUP) is a simplified, functional version of a product or solution that allows users to interact with it and provide feedback.

Key Points:

- Focuses on core features, not full functionality
- Quick to build and test
- Helps identify design issues early

- Reduces risk of investing in unnecessary features

2.***Difference Between MUP, MVP, and Mockups***

Prototype Type	Description	Purpose	Example
MUP (Minimum Usable Prototype)	Functional, simple version that users can actually use	Test usability and core features	A basic mobile app that lets students check classroom availability and book a room, without notifications or fancy design
MVP (Minimum Viable Product)	Slightly more complete than MUP, ready for market testing	Test market acceptance and basic functionality	A mobile app with classroom booking, notifications, and simple dashboard, released to all students for feedback
Mockup	Visual representation or design sketch	Test appearance or layout without usability	A static design showing the app screens with buttons, colors, and layout, but users cannot interact with it

"MUP is usable by users, unlike mockups which are just visual designs. It's also quicker and simpler than an MVP."

3. Why MUP is Important

1. **Early User Feedback:**
   - ✓ Users interact with a real version and provide feedback
2. **Saves Time and Cost:**
   - ✓ Identify issues before building the full product
3. **Validates Core Concept:**
   - ✓ Confirms that the main idea solves the user's problem

4. **Encourages Iteration:**
    - ✓ Supports quick changes based on testing

4.Steps to Build a Minimum Usable Prototype

1. **Identify Core Features:**
    - ✓ Focus only on what solves the user's problem
2. **Sketch or Wireframe:**
    - ✓ Visualize the design before building
3. **Build Prototype:**
    - ✓ Can be digital (app, website) or physical (model, cardboard, paper)
4. **Test with Users:**
    - ✓ Observe usage, gather feedback, note issues
5. **Iterate:**
    - ✓ Refine based on feedback

5.Example – Gym/Fitness Center Scenario

Problem: Gym members waste time trying to find available exercise machines during peak hours

MUP Solution:

- A simple mobile app that shows which machines are currently free in real-time
- **Core feature:** Display a list of machines and their availability
- Users can test by checking machine status before going to the gym and giving feedback

Next Steps:

1. **Collect feedback on usability:**
    - ✓ Can users easily see which machines are free?

- ✓ Is the information accurate and updated in real-time?

2. **Identify missing features or confusing elements:**
   - ✓ Would users like to reserve machines in advance?
   - ✓ Are notifications about machine availability useful?
3. **Iterate improvements:**
   - ✓ Add a "favorite machines" feature
   - ✓ Include estimated wait times for busy machines

"*Even a very simple app that only shows available machines can be a MUP. The purpose is to* ***test usability and gather user insights****, not to build a fully-featured product immediately.*"

6.Classroom Activity

Activity: Create a Paper MUP

Steps:

1. Form small groups
2. Select a validated problem from previous units
3. Identify core features for the MUP
4. Build a paper or cardboard prototype, digital sketch, or clickable wireframe
5. Test the prototype within the class and collect feedback

Purpose:

- Hands-on experience in rapid prototyping
- Reinforces focus on usability and learning

7.Common Mistakes

☐ Trying to build a full product instead of a minimal version
☐ Ignoring user feedback
☐ Overcomplicating the design

*"The purpose of a MUP is **learning quickly and cheaply**, not creating a polished product."*

8.Key Takeaways

- **MUP =** Minimum Usable Prototype
- Focuses on core features and usability
- Helps validate ideas and gather feedback early
- Encourages iterative learning before building full solutions

---

Period 2

MUP Challenge Brief

"MUP Challenge Brief = A structured guideline for designing a Minimum Usable Prototype to test and validate a solution."

1. What is a MUP Challenge Brief?

Definition:

A MUP Challenge Brief is a concise, well-defined problem statement with objectives, constraints, and expected outcomes, designed to guide teams in building and testing a Minimum Usable Prototype.

Key Points:

- Provides a clear problem context
- Sets boundaries and goals for the prototype
- Encourages creative and focused solutions
- Ensures students test real user needs

2.Components of a MUP Challenge Brief

Component	Description	Example
Problem Statement	A clear description of the problem to solve	Students waste time finding empty classrooms
Target Users	Who the solution is for	College students needing study spaces
Core Needs	Main needs to address	Find available classrooms quickly
Constraints	Limitations for prototype	Build within 1 hour using paper/digital tools
Success Criteria	How to evaluate MUP	Users can locate a room within 2 minutes
Deliverables	Expected output from teams	Paper prototype or wireframe showing core functionality

"Each component ensures that the prototype is **focused, usable, and testable**."

3.Why Use a MUP Challenge Brief

1. **Focus on the Core Problem:**
   - ✓ Avoids unnecessary features and complexity
2. **Encourages Creativity:**
   - ✓ Students can explore innovative solutions within constraints
3. **Provides Clear Evaluation:**
   - ✓ Helps instructors and peers assess usability and effectiveness
4. **Promotes Team Collaboration:**
   - ✓ Teams work together to understand the problem and design a solution

4. Example – MUP Challenge Brief

Scenario: Students miss important college announcements

Challenge Brief:

- **Problem Statement:** Students often miss important notices about class changes, events, or deadlines because information is scattered across notice boards and messages.
- **Target Users:** College students from all departments
- **Core Needs:** View all important announcements quickly in one place
- **Constraints:** Prototype should be built in 1 hour using paper sketches or simple digital tools; no complex coding required
- **Success Criteria:** Users can find the latest important announcement within 1 minute using the prototype
- **Deliverables:** A paper or digital prototype showing core functionality (announcement list, priority tags, or simple notifications)

"This brief clearly sets expectations by defining the problem, users, constraints, and success criteria, helping the team focus on a simple, testable solution."

5.Classroom Activity

Activity: Build MUP Using Challenge Brief

Steps:

1. Form small groups
2. Assign the MUP Challenge Brief (or allow groups to create their own based on validated problems)
3. Identify core features for the prototype
4. Build the Minimum Usable Prototype (paper, cardboard, or digital wireframe)
5. Test the MUP within the class and collect feedback from peers

Purpose:

- Hands-on experience in rapid prototyping
- Reinforces teamwork, problem-solving, and user-centric design

6.Best Practices

- Focus on **core** functionality first
- Keep the prototype simple and testable
- Observe peer feedback carefully
- Iterate quickly based on insights

7.Common Mistakes

☐ Trying to build a full product instead of a minimal version
☐ Ignoring the problem constraints or success criteria
☐ Neglecting user feedback during testing

8. Key Takeaways

- A MUP Challenge Brief provides structure and clarity for prototype development
- Focuses on core features, usability, and validation
- Encourages teamwork, creativity, and iterative learning
- Essential step before moving to full product design

Period 3

Designing & Crafting the Value Proposition

"Value Proposition = The benefits a product or service delivers to the user and why it is better than alternatives."

1. What is a Value Proposition?

Definition:

A Value Proposition is a clear statement that explains how a product or service solves a user problem, delivers benefits, and differentiates itself from alternatives.

Key Points:

- Focuses on user benefits, not features
- Answers the questions:
    1. What problem does the product solve?
    2. Who is the target user?
    3. Why is this solution better than alternatives?

2.Components of a Value Proposition

Component	Description	Example
Target User	Who the solution is for	Undergraduate students needing study spaces
Problem	The main pain point	Wasting time finding empty classrooms
Solution	How the product solves the problem	Mobile app showing available classrooms in real-time
Benefits	What the user gains	Saves 20–30 minutes, reduces frustration
Differentiation	Why it's better than alternatives	Combines map view and notifications, unlike manual search

"A strong value proposition clearly communicates the benefits and sets your solution apart."

3.Why Designing a Value Proposition is Important

1. **Clarifies Purpose:**
   - ✓ Helps the team understand the core value of the solution
2. **Communicates with Users:**
   - ✓ Explains why users should choose your solution
3. **Guides Design Decisions:**
   - ✓ Ensures features align with user benefits
4. **Supports Prototyping and Testing:**
   - ✓ Provides a benchmark to test MUPs and solutions

4.Steps to Craft a Value Proposition

1. **Understand the User:**
   - ✓ Use user personas and stories
2. **Identify Core Problems:**
   - ✓ Focus on the highest significance and incidence problems
3. **Define Key Benefits:**
   - ✓ How the solution reduces pain or adds value
4. **Compare Alternatives:**
   - ✓ Identify what users currently do to solve the problem
5. **Write the Value Proposition Statement:**
   - ✓ Example template:

> "For [target user], who [problem], our [solution] provides [benefit] unlike [alternative]."

5.Example – Value Proposition Statement

Problem: Job seekers struggle to track multiple job applications

Value Proposition Statement:

"For job seekers who apply to many positions and lose track of responses, our application-tracking platform organizes all applications,

deadlines, and follow-ups in one place, reducing stress and missed opportunities, unlike scattered emails and spreadsheets."

"This statement clearly highlights the target user, problem, solution, benefits, and differentiation."

6.Classroom Activity

Activity: Craft Value Propositions

Steps:

1. Form small groups
2. Each group selects a validated problem and target user/persona
3. Use the template to craft a value proposition statement
4. Share statements with the class and discuss:
    - ✓ Is it clear and concise?
    - ✓ Does it focus on user benefits?
    - ✓ Does it differentiate from existing solutions?

Purpose:

- Students learn to translate insights into clear value
- Reinforces user-centered thinking and communication skills

7. Common Mistakes

☐ Focusing on features instead of benefits
☐ Being vague or generic
☐ Ignoring differentiation from alternatives

"A value proposition should clearly answer why a user should care about your solution."

8.Key Takeaways

- Value Proposition = What the product offers, why it matters, and how it differs
- Essential for aligning design with user needs
- Supports prototyping, testing, and pitching solutions
- Provides a shared understanding for the team and users

Period 4

Designing the Value Proposition

"Designing a Value Proposition = Creating a clear, focused, and compelling statement that highlights how the solution benefits the user."

1. What is Designing a Value Proposition?

Definition:

Designing a Value Proposition is the process of formulating, refining, and visualizing the benefits your solution provides to target users, ensuring it is compelling and clearly differentiates from alternatives.

Key Points:

- Focus on user needs and benefits
- Incorporate insights from user research, personas, and stories
- Helps prioritize features and guide prototype development

2.Steps to Design a Value Proposition

1. **Understand User Needs:**
    - ✓ Use user personas and validated problems

2. **Identify Key Benefits:**
    - ✓ Determine how the solution solves problems or adds value
3. **Evaluate Alternatives:**
    - ✓ Compare with current solutions to highlight differentiation
4. **Create the Value Proposition Statement:**
    - ✓ Template:

> "For [target user], who [problem], our [solution] provides [benefit] unlike [alternative]."

5. **Visualize the Proposition (Optional):**
    - ✓ Use diagrams, charts, or value proposition canvases to make it clear
6. **Validate and Iterate:**
    - ✓ Test with users to ensure clarity, relevance, and appeal
    - ✓ Refine based on feedback and insights

3.Example – Value Proposition Design Process

Problem: People forget to take their daily medicines on time

1. **Target User:** Elderly people and patients with long-term medication
2. **Key Benefit:** Take medicines on time, avoid missed doses
3. **Alternative Solutions:** Paper reminders, alarm clocks, caregiver reminders
4. **Value Proposition Statement:**

"For patients who often forget to take their medicines, our smart medication reminder app sends timely alerts and tracks doses, helping them stay healthy and worry-free, unlike paper notes or basic alarms."

"This value proposition clearly identifies the user, the problem, the main benefit, and explains how the solution is better than existing alternatives."

4.Classroom Activity

Activity: Design a Value Proposition

Steps:

1. Form small groups
2. Choose a validated problem and target user/persona
3. Identify core benefits and differentiation points
4. Draft a value proposition statement using the template
5. Present to the class and receive peer feedback
6. Refine statements based on feedback

Purpose:

- Students practice translating user insights into compelling solutions
- Encourages clarity, focus, and user-centered thinking

5. Best Practices

- Focus on benefits, not features
- Be specific and concise
- Highlight differentiation from alternatives
- Use simple, user-friendly language

6.Common Mistakes

☐ Focusing on what the product does instead of how it benefits users

☐ Being vague or generic

☐ Ignoring the alternatives users currently use

"A good value proposition must clearly communicate why the solution matters to the user and why it is better than what they already do."

7.Key Takeaways

- Designing a value proposition ensures user-centered clarity and focus
- Guides feature prioritization and prototype design
- Helps communicate solution value to stakeholders and users
- Forms the foundation for testing and iteration

Period 5

Testing the Value Proposition

"*Testing the Value Proposition = Validating with real users whether your solution's benefits are clear, relevant, and compelling.***"**

***1.What is Testing the Value Proposition*?**

Definition:

Testing the Value Proposition is the process of evaluating your value proposition with target users to ensure it resonates, solves their problem, and motivates them to adopt the solution.

Key Points:

- Helps confirm user interest and perceived value
- Reduces risk of building solutions that users do not want
- Provides insights for iteration and improvement

2. Why Test the Value Proposition

1. **Validate User Needs:**
    - ✓ Ensure your solution addresses real pain points

2. **Clarify Benefits:**
    - ✓ Check if the stated benefits are clear and relevant
3. **Reduce Risk:**
    - ✓ Avoid investing in solutions that users may not adopt
4. **Improve Communication:**
    - ✓ Helps refine messaging to make value clear and compelling

3.Methods to Test the Value Proposition

Method	Description	Example
Customer Interviews	Ask users directly if the solution and benefits make sense	"Would this app help you find empty classrooms faster?"
Surveys / Questionnaires	Collect quantitative feedback from multiple users	"Rate how useful this solution would be on a scale of 1–5"
Prototype Testing (MUP)	Observe users interacting with a simple prototype	Use a paper or digital prototype of the classroom booking app
A/B Testing	Compare multiple value proposition messages	Test two versions of the benefit statement: "Find classrooms fast" vs. "Save 20–30 mins daily"

"The goal is to gather evidence and feedback that shows users find your value proposition relevant and compelling."

4.Steps to Test the Value Proposition

1. **Select Target Users:**
    - ✓ Use validated personas from previous units
2. **Prepare Questions or Scenarios:**
    - ✓ Ask about clarity, relevance, and attractiveness of benefits
3. **Present the Value Proposition:**
    - ✓ Can be verbally, visually, or via MUP

4. **Collect Feedback:**
    - ✓ Listen, observe reactions, and take notes
5. **Analyze Results:**
    - ✓ Identify if users understand the benefits and see value
6. **Iterate:**
    - ✓ Refine the value proposition based on insights and retest if necessary

5.Example – Value Proposition Testing Scenario

Problem: People forget to pay utility bills on time

Value Proposition:
"For working adults who forget bill due dates, our bill-reminder app sends timely alerts and payment summaries, helping them avoid late fees and stress."

Testing Process:

- Conduct interviews with 10 working adults
- Ask:
    - ✓ "Does this solution solve your problem?"
    - ✓ "Which benefit matters most to you?"
    - ✓ "Would reminders alone be enough, or do you want payment links?"
- **Observations:** Users strongly value avoiding late fees
- **Feedback:** Many request direct payment links inside the app

Outcome:
Refine the value proposition to highlight late-fee prevention and one-click payments**.**

6.Classroom Activity

Activity: Test Your Value Proposition

Steps:

1. Form small groups
2. Use your designed value proposition from the previous period
3. Conduct mini-tests with peers:
    - ✓ Ask questions about clarity, relevance, and appeal
4. Collect feedback and discuss improvements
5. Revise the value proposition based on feedback

Purpose:

- Provides hands-on experience in validation
- Reinforces user-centered iteration
- Helps build confidence in communicating value

7.Best Practices

- Test with real or representative users
- Ask open-ended questions to understand user perspective
- Observe non-verbal cues and reactions
- Be ready to iterate multiple times

8. Common Mistakes

 Testing only with friends or biased users

☐ Ignoring feedback and sticking to assumptions

☐ Focusing on features instead of benefits

9.Key Takeaways

- Testing the value proposition ensures solution relevance and appeal

- Helps identify misunderstandings or missing benefits
- Feedback guides iteration and improvement before full development
- Integral part of user-centered innovation and applied design thinking

---

Period 6

Design a Compelling Value Proposition

"Compelling Value Proposition = A clear, user-centered statement that communicates the unique benefits and motivates adoption."

1.What Makes a Value Proposition Compelling?

Definition:

A compelling value proposition is a user-centered statement that clearly communicates the problem being solved, the benefits delivered, and why the solution is better than available alternatives in a way that engages and motivates the user.

Key Attributes:

1. Clarity: Users immediately understand the benefit
2. Relevance: Addresses a real and important user problem
3. Differentiation: Explains why it is better than other solutions
4. Emotional Appeal: Connects with users' pain points, desires, or goals
5. Conciseness: Communicated in simple and few words

2. Steps to Design a Compelling Value Proposition

1. **Know Your Users Deeply:**
   - ✓ Use **validated personas and stories**

2. **Focus on Key Benefits:**
    - ✓ Prioritize benefits that are **most important to users**
3. **Highlight Differentiation:**
    - ✓ Clearly show **why your solution is unique**
4. **Use Clear, Simple Language:**
    - ✓ Avoid jargon or complex descriptions
5. **Incorporate Emotion and Motivation:**
    - ✓ Show **how the solution improves the user's life or experience**
6. **Iterate and Refine:**
    - ✓ Test the statement with users and improve clarity, relevance, and appeal

3.Example – Healthcare Appointment Scenario

Problem: Patients wait too long to book doctor appointments

Compelling Value Proposition:
"For busy patients who struggle to get timely doctor appointments, our online scheduling platform shows real-time doctor availability and enables instant booking, reducing wait times and improving access to healthcare compared to phone-based systems."

Analysis of Compelling Attributes

- **Clarity:** Clearly explains what the solution offers (real-time doctor availability and instant booking)
- **Relevance:** Solves a common and important problem (long waiting times for appointments)
- **Differentiation:** Highlights improvement over traditional methods (phone calls and manual scheduling)
- **Emotional Appeal:** Reduces frustration and anxiety related to delayed medical care

- **Conciseness:** Communicates the value in a single, focused sentence

4. Classroom Activity

Activity: Design a Compelling Value Proposition

Steps:

1. Form small groups
2. Choose a **validated problem and user persona**
3. Draft a **compelling value proposition** using the template:
   - ✓ "For [target user] who [problem], our [solution] provides [benefit] unlike [alternative]."
4. Refine it to be **clear, relevant, differentiated, and engaging**
5. Share with the class for **feedback and suggestions**

Purpose:

- Students practice creating **value propositions that motivate adoption**
- Reinforces **clarity, focus, and user-centered thinking**

5.Best Practices

- Focus on **benefits, not features**
- Highlight **what makes the solution unique**
- Keep the message **simple and memorable**
- Test with users to ensure **clarity and engagement**
- Use **emotional and practical appeal**

6. Common Mistakes

 Making it too long or complicated

☐ Focusing only on features rather than benefits

☐ Ignoring differentiation or emotional appeal

☐ Assuming users will automatically understand the value

7.Key Takeaways

- A compelling value proposition **clearly communicates benefits, differentiates the solution, and motivates adoption**
- Critical for **customer engagement and prototype testing**
- Supports **team alignment and focused solution development**
- Should be **iterative and validated with real users**

Period 7

Process, Tools, and Techniques of Value Proposition Design

"Value Proposition Design = Structured process + tools + techniques to create user-centered, compelling solutions."

1.The Process of Value Proposition Design

1. **Understand Your Customer**
   - ✓ Research user needs, pain points, desires
   - ✓ Use **personas, user stories, and field observations**
2. **Map Customer Jobs, Pains, and Gains**
   - ✓ Identify what the user is **trying to achieve (jobs)**
   - ✓ Recognize **pain points and obstacles**
   - ✓ Determine **gains or desired outcomes**
3. **Design Solutions to Address Jobs and Pains**
   - ✓ Focus on features that **solve key problems**
   - ✓ Emphasize **benefits that deliver value**

4. **Craft the Value Proposition**
    - ✓ Combine solution benefits with user needs
    - ✓ Highlight **differentiation and emotional appeal**
5. **Validate with Users**
    - ✓ Use **MUPs, prototypes, and user testing**
    - ✓ Gather feedback and iterate
6. **Refine and Finalize**
    - ✓ Adjust messaging, benefits, or features based on insights
    - ✓ Ensure clarity, relevance, and engagement

*"This process ensures your value proposition is **user-centered, compelling, and validated before full-scale development**."*

2.Tools for Value Proposition Design

Tool	Purpose	Example
Value Proposition Canvas	Maps customer jobs, pains, and gains against solution benefits	Helps identify key benefits for students finding classrooms
Persona Profiles	Represent target users with demographics, goals, and pain points	"Rohan, 20-year-old student, struggles to find study spaces"
User Stories	Short descriptions of what a user wants to achieve	"As a student, I want to know which classrooms are free so I can study efficiently"
MUP / Prototype	Test core solution features	Simple mobile app showing classroom availability
Customer Interviews /	Gather feedback and validate the value	Ask students if the app saves time and reduces

Tool	Purpose	Example
Surveys	proposition	stress

3.Techniques for Value Proposition Design

1. **Empathy Mapping**
   - ✓ Understand what users **think, feel, say, and do**
   - ✓ Identify emotional drivers and obstacles
2. **Jobs-to-be-Done Analysis**
   - ✓ Identify **functional, social, and emotional jobs** users want to complete
3. **Pain-Gain Mapping**
   - ✓ Align **pain points with solutions** and highlight potential gains
4. **Rapid Prototyping**
   - ✓ Build quick **MUPs or sketches** to test value propositions
5. **Iteration & Feedback Loops**
   - ✓ Continuously refine value proposition based on **user feedback**

4. Example – Working Professional Scenario

Problem: Employees forget or miss important task deadlines

Process:

1. **Customer Jobs:** Track tasks and meet deadlines efficiently
2. **Pains:** Missed deadlines, work stress, poor performance reviews
3. **Gains:** Stay organized, meet deadlines, improve productivity
4. **Value Proposition Statement:**
   "For working professionals who struggle to keep track of multiple deadlines, our smart task management app provides automated reminders and priority-based task organization, helping them stay

productive and reduce work-related stress, unlike basic to-do lists or manual tracking."

Tools Used:

- **Value Proposition Canvas**
- **Persona:** "Anita" (Corporate employee)
- **User story and MVP** for usability testing

Technique:

- **Empathy mapping** to understand workload pressure and stress
- **Rapid prototyping** tested with office professionals to refine reminders and task prioritization

5.Classroom Activity

Activity: Apply Value Proposition Design Tools

Steps:

1. Form small groups
2. Select a **validated problem and user persona**
3. Use **Value Proposition Canvas** to map jobs, pains, and gains
4. Draft a **value proposition statement**
5. Share and refine based on **peer feedback**

Purpose:

- Provides hands-on experience with **tools and techniques**
- Encourages **structured thinking and iteration**
- Reinforces **user-centered design principles**

6.Best Practices

- Start with **deep user understanding**
- Focus on **high-impact problems and benefits**

- Use **visual tools** to clarify ideas
- Iterate frequently based on **user insights**

7.Common Mistakes

 Skipping user research and assumptions

☐ Ignoring emotional aspects of user needs

☐ Overloading value proposition with too many features

8.Key Takeaways

- **Value Proposition Design** is a structured process using tools and techniques
- Focuses on **user jobs, pains, gains, and benefits**
- Combines **design thinking with validation**
- Ensures solutions are **user-centered, compelling, and testable**

Period 8

Tools of Value Proposition Design

"Tools = Visual and practical aids that help design, test, and communicate a value proposition clearly and effectively."

1.*Importance of Tools in Value Proposition Design*

- Tools provide **structure** for creating user-centered value propositions
- Make **complex ideas simple and visual**
- Facilitate **team collaboration and communication**
- Enable **testing and iteration** before building the full solution

2.Key Tools of Value Proposition Design

Tool	Purpose	Example
Value Proposition Canvas (VPC)	Maps user **jobs, pains, and gains** against product **pain relievers and gain creators**	Helps design a student classroom app that solves time-wasting problems
Empathy Map	Understand what users **think, feel, say, and do**	Identify students' frustrations during campus navigation
Customer Persona Profiles	Represent typical users with **demographics, behaviors, needs**	"Rohan – 20-year-old student struggling to find study spaces"
User Stories	Capture **specific user goals and needs**	"As a student, I want to know which classrooms are free so I can study efficiently"
Prototypes / MUPs	Test value proposition in **real usage scenarios**	Paper or digital prototype showing available classrooms
Customer Interviews & Surveys	Collect **qualitative and quantitative feedback**	Ask students how valuable the app features are
Journey Mapping	Visualizes **user experience across touchpoints**	Shows how a student searches, finds, and books a classroom
A/B Testing / Experiments	Compare **different value proposition messages**	"Save time" vs. "Reduce stress" messaging for classroom app

3. How to Use These Tools Effectively

1. **Start with User Research**
   - ✓ Collect insights using **interviews, surveys, and observation**
2. **Visualize User Needs**
   - ✓ Use **Empathy Maps, Personas, and Journey Maps**

3. **Align Product Features with User Benefits**
    - ✓ Use the **Value Proposition Canvas** to match **pain relievers and gain creators**
4. **Capture User Goals and Stories**
    - ✓ Translate needs into **actionable user stories**
5. **Prototype and Test**
    - ✓ Build **MUPs** and validate value propositions with real users
6. **Iterate Based on Feedback**
    - ✓ Refine value propositions and solutions continuously

4.Example – Restaurant Queue Management Scenario

Problem: Customers waste time waiting in long restaurant queues

Application of Tools:

1. **Persona:** Meera – Working professional, prefers quick dining during lunch breaks
2. **Empathy Map:** Shows impatience, hunger, and frustration caused by uncertain wait times
3. **Value Proposition Canvas:** Maps Meera's pains (long waits, unclear queue status) against solution benefits (digital queue, wait-time alerts)
4. **User Story:**
   "As a customer, I want to know my waiting time and reserve a spot in line so I can avoid standing in queues."
5. **Prototype / MVP:** Mobile/web app mock-up showing live queue position and estimated seating time
6. **Customer Interview:** Test whether digital queue tracking reduces waiting stress and improves dining experience

5.Classroom Activity

Activity: Apply Value Proposition Tools

Steps:

1. Form small groups
2. Select a **validated problem and user persona**
3. Use at least **3 tools** (Persona, Value Proposition Canvas, User Story) to design a solution
4. Build a **mini MUP** for testing
5. Share and discuss outcomes with the class

Purpose:

- Provides hands-on experience using **practical tools**
- Reinforces **structured thinking and user-centered design**
- Encourages **team collaboration and iteration**

6.Best Practices

- Start with **user research** to guide tool usage
- Combine multiple tools for **holistic understanding**
- Visualize ideas clearly for **team and user comprehension**
- Use prototypes to **validate assumptions early**

7.Common Mistakes

☐ Using tools superficially without deep insights
☐ Focusing on features instead of **user benefits**
☐ Ignoring iteration and user feedback

8.Key Takeaways

- Tools are essential for **designing, visualizing, and validating value propositions**

- Enable **team alignment, user understanding, and iterative learning**
- Include **Value Proposition Canvas, Empathy Maps, Personas, User Stories, Prototypes, and Journey Maps**
- Support **customer-centric innovation and applied design thinking**

Period 9

Techniques of Value Proposition Design

"Techniques = Methods and approaches to systematically create and improve value propositions based on user needs and feedback."

1.Importance of Techniques in Value Proposition Design

- Techniques provide **systematic approaches** to uncover user needs and craft solutions
- Enable **insight-driven innovation**
- Help in **prioritizing benefits and features**
- Facilitate **testing and iteration**

2.Key Techniques of Value Proposition Design

Technique	Purpose	Example
Empathy Mapping	Understand what users **think, feel, say, and do**	Map student frustrations while finding classrooms
Customer Interviews	Gather **qualitative insights** from real users	Ask students which solution features are most valuable
Jobs-to-be-Done (JTBD)	Identify **functional, emotional, and social jobs** users want to achieve	Students want to "study efficiently without wasting time"

Technique	Purpose	Example
Pain-Gain Analysis	Identify **pain points** and potential **value gains**	Pain: wasted time; Gain: save 20–30 minutes per day
Rapid Prototyping / MUP	Build **quick, testable solutions**	Paper or digital prototype of classroom availability app
Storyboarding / Journey Mapping	Visualize **user interactions and touchpoints**	Show how a student searches, finds, and books a classroom
A/B Testing	Test **alternative messages or designs** to identify most compelling value proposition	Test "Save time" vs. "Reduce stress" messaging for the app
Iterative Feedback Loops	Continuously refine the value proposition based on **user input**	Update prototype and messaging based on student feedback

3.Steps to Apply Techniques Effectively

1. **Start with User Research:**
   - ✓ Gather insights using **interviews, surveys, and observations**
2. **Analyze User Needs and Problems:**
   - ✓ Use **Empathy Maps, JTBD, and Pain-Gain Analysis**
3. **Visualize the Solution:**
   - ✓ Use **Storyboarding, Journey Mapping, or Value Proposition Canvas**
4. **Prototype and Test:**
   - ✓ Build a **MUP** and test with real users
5. **Collect Feedback and Iterate:**
   - ✓ Refine the value proposition based on insights
   - ✓ Repeat testing until **clear, relevant, and compelling**

4.Example – Smart Parking Scenario

Problem: Drivers waste time searching for available parking spaces

Application of Techniques:

1. **Empathy Mapping:** Identify driver frustration, anxiety, and urgency when parking
2. **JTBD:**
   - ✓ *Functional job:* Find a nearby parking spot quickly
   - ✓ *Emotional job:* Reduce stress and avoid being late
3. **Pain–Gain Analysis:**
   - ✓ *Pain:* Circling for 15–25 minutes, fuel waste, irritation
   - ✓ *Gain:* Faster parking, cost savings, peace of mind
4. **Prototyping / MVP:** Mobile app showing real-time available parking spots
5. **Storyboarding:** Map how a driver enters an area, checks the app, navigates to a spot, and parks
6. **A/B Testing:** Test different alert messages (e.g., urgency vs. reassurance)
7. **Iterative Feedback:** Improve accuracy, alerts, and UI based on driver testing

Outcome: A validated, user-centered value proposition that saves time, reduces stress, and improves the urban driving experience.

5.Classroom Activity

Activity: Apply Techniques to Value Proposition Design

Steps:

1. Form small groups

2. Select a **validated problem and persona**
3. Apply **at least 3 techniques** (e.g., Empathy Map, JTBD, MUP) to create or refine a value proposition
4. Test mini prototypes with peers and gather feedback
5. Refine value propositions based on feedback

Purpose:

- Hands-on experience with **practical design techniques**
- Reinforces **user-centered thinking, iteration, and testing**
- Builds confidence in **creating compelling solutions**

6. Best Practices

- Combine multiple techniques for **holistic understanding**
- Start with **user needs and pain points**
- Prototype early and gather **real user feedback**
- Iterate continuously to **refine value propositions**

7.Common Mistakes

 Using techniques in isolation without integration
☐ Ignoring emotional and social aspects of user needs
☐ Failing to iterate based on feedback
☐ Overcomplicating the process instead of focusing on core benefits

8.Key Takeaways

- Techniques are **practical approaches to design, test, and refine value propositions**
- Include **Empathy Mapping, JTBD, Pain-Gain Analysis, Storyboarding, Prototyping, and A/B Testing**
- Ensure solutions are **user-centered, relevant, and compelling**

- Support **iterative, evidence-based applied design thinking**

UNIT IV

CONCEPT GENERATION

Period 1

Solution Exploration

"Solution Exploration = Investigating multiple solution ideas to identify the most feasible, valuable, and usable concept."

1.What is Solution Exploration?

Definition:

Solution Exploration is the process of **generating, investigating, and analyzing multiple potential solutions** to a validated problem to identify the most promising concept for prototyping.

Key Points:

- Encourages **divergent thinking** – exploring many ideas
- Focuses on **user needs, value, and feasibility**
- Reduces risk of **selecting the wrong solution too early**
- Supports **iterative learning and refinement**

2.Steps in Solution Exploration

1. **Review Problem and User Needs:**
   - ✓ Understand the **validated problem, personas, and user stories**
2. **Brainstorm Multiple Solutions:**
   - ✓ Generate **as many ideas as possible** without judging them
   - ✓ Encourage **creative and out-of-the-box thinking**

3. **Shortlist Promising Concepts:**
    - ✓ Evaluate ideas for **feasibility, usability, and value**
4. **Prototype and Test Ideas (MUPs):**
    - ✓ Build simple, testable versions of shortlisted solutions
5. **Iterate Based on Feedback:**
    - ✓ Refine concepts using **user feedback and observation**

3. Why Solution Exploration is Important

1. **Encourages Creativity:**
    - ✓ Multiple ideas lead to **innovative solutions**
2. **Reduces Risk:**
    - ✓ Early evaluation prevents **investing in ineffective solutions**
3. **Supports User-Centric Design:**
    - ✓ Focuses on **meeting real user needs**
4. **Enables Iterative Learning:**
    - ✓ Students **explore, test, and refine** continuously

4. Techniques for Solution Exploration

Technique	Description	Example
Brainstorming	Generate multiple ideas without judgment	Sketching 10+ ways to show classroom availability
Mind Mapping	Visually organize ideas and connections	Map ideas like notifications, map view, or booking system
SCAMPER Method	Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Reverse	Adapt a booking system from library seats for classrooms
Sketching & Storyboarding	Visualize ideas quickly	Draw sequences of how students find and book

Technique	Description	Example
		classrooms
Rapid Prototyping (MUP)	Build simple models to test ideas	Paper prototype of an app showing classroom availability

5.Example – Library Seat Availability Scenario

Problem: Library users waste time searching for empty study seats

Solution Exploration Ideas:

1. **Mobile app** showing real-time seat availability by section
2. **Sensor-based indicators** (green/red lights) on study tables
3. **Digital display boards** at library entrances showing free seats
4. **Seat reservation system** with time limits
5. **QR-code check-in/check-out** system for users to update seat status

Shortlisting Concepts:

- Evaluate based on **accuracy, user effort, cost, scalability, and maintenance**
- Select the **top 2–3 solutions** (e.g., app + digital display) to prototype as MVPs

6.Classroom Activity

Activity: Explore Solution Concepts

Steps:

1. Form small groups
2. Review a **validated problem and persona**
3. **Brainstorm at least 5 solution ideas**
4. Use **sketching or mind mapping** to visualize concepts
5. Shortlist **2–3 promising ideas** for MUP development

Purpose:

- Encourage **creative exploration**
- Practice **rapid idea generation and evaluation**
- Prepare for **MUP prototyping in subsequent periods**

7.Best Practices

- Focus on **quantity first, quality later** during brainstorming
- Use **visualization tools** like sketches and storyboards
- Keep the **user's needs and pain points central**
- Be open to **iteration and combination of ideas**

8.Common Mistakes

☐ Selecting the first idea without exploration
☐ Ignoring user needs and feasibility
☐ Being overly critical during brainstorming
☐ Failing to iterate based on feedback

9.Key Takeaways

- Solution Exploration is about **diverging to generate ideas and converging to select the best ones**
- Encourages **creativity, iteration, and user-centric thinking**
- Uses techniques like **brainstorming, mind mapping, SCAMPER, sketching, and MUPs**
- Forms the foundation for **concept generation and prototyping**

Period 2

Concepts Generation

"Concepts Generation = Systematically transforming multiple ideas into structured, feasible, and testable solution concepts."

1.What is Concepts Generation?

Definition:

Concepts Generation is the process of **taking ideas from solution exploration and systematically developing them into concrete, actionable concepts** that can be prototyped and tested.

Key Points:

- Moves from **divergent thinking (many ideas)** to **convergent thinking (selecting best ideas)**
- Focuses on **user value, feasibility, and usability**
- Ensures solutions are **practical and ready for prototype development**

2.Steps in Concepts Generation

1. **Collect and Review Ideas:**
   - ✓ Gather all solution ideas from brainstorming or exploration
2. **Group Similar Ideas:**
   - ✓ Organize ideas into categories or themes
3. **Evaluate Feasibility and Impact:**
   - ✓ Consider **technical feasibility, usability, cost, and value to users**
4. **Refine and Combine Concepts:**
   - ✓ Merge complementary ideas for stronger concepts

5. **Select Promising Concepts:**
    - ✓ Choose **2–3 concepts** to develop as MUPs
6. **Document Concept Details:**
    - ✓ Include problem addressed, target users, key benefits, and possible features

3.Why Concepts Generation is Important

1. **Prioritizes Best Ideas:**
    - ✓ Ensures only **high-value, feasible solutions** move forward
2. **Reduces Risk:**
    - ✓ Avoids wasting time and resources on **impractical ideas**
3. **Supports Prototyping:**
    - ✓ Provides **well-defined concepts ready for MUP design**
4. **Encourages Iteration:**
    - ✓ Concepts can be **refined and improved** based on feedback

4.Techniques for Concepts Generation

Technique	Description	Example
Affinity Diagramming	Group similar ideas to identify patterns	Combine app features like notifications and map view into one concept
Multi-Criteria Evaluation	Evaluate ideas based on **feasibility, value, usability, and cost**	Score ideas 1–5 for each criterion to select the best
Concept Sketching / Storyboarding	Visualize how concepts would work	Draw sequences showing how a student finds and books a classroom
Concept Combination	Merge complementary ideas into a stronger concept	Merge digital notice board and mobile notifications into one solution
Dot Voting /	Team votes to identify	Students vote on top 2

Technique	Description	Example
Prioritization	the most promising concepts	concepts for prototyping

5.Example – Online Course Assignment Feedback Scenario

Problem:

Students feel confused and discouraged because assignment feedback is unclear and delayed.

Generated Concepts:

1. **Instant auto-feedback system**
   - ✓ Gives basic feedback immediately after submission
   - ✓ Helps students understand mistakes early
2. **Clear rubric-based feedback dashboard**
   - ✓ Shows marks and comments for each evaluation criterion
   - ✓ Makes grading transparent and easy to understand
3. **Audio feedback from instructors**
   - ✓ Teachers record short voice notes
   - ✓ Feedback feels more personal and encouraging
4. **Peer feedback tool**
   - ✓ Students review each other's work using guided questions
   - ✓ Encourages learning from peers

Concept Selection:

- Concepts are evaluated using **multiple criteria**:
  Educational value, clarity, ease of use, instructor effort, and cost
- **Top 2 concepts selected for MVP prototyping:**
  - ✓ Instant auto-feedback system
  - ✓ Rubric-based feedback dashboard

Refinement:

- Combine **instant feedback** with **clear rubrics** into **one simple solution**
- Students receive immediate guidance and detailed evaluation in one place

6. Classroom Activity

Activity: Generate and Select Concepts

Steps:

1. Form small groups
2. Review ideas from **Solution Exploration**
3. Use **Affinity Diagramming or Multi-Criteria Evaluation** to organize and assess ideas
4. Combine complementary ideas to strengthen concepts
5. Select **2–3 promising concepts** for MUP development

Purpose:

- Reinforces **systematic idea evaluation**
- Encourages **collaboration and critical thinking**
- Prepares concepts for **prototype design in upcoming periods**

7.Best Practices

- Evaluate ideas **objectively using criteria**
- Encourage **creative combinations of ideas**
- Keep **user needs central** in concept development
- Document concepts clearly for **prototype development**

8.Common Mistakes

☐ Selecting ideas based on personal preference instead of user value
☐ Ignoring feasibility or technical limitations
☐ Failing to refine or combine complementary ideas
☐ Skipping documentation, which makes prototyping harder

9. Key Takeaways

- Concepts Generation is the **structured transition from ideas to actionable concepts**
- Focuses on **feasibility, usability, and value to users**
- Uses techniques like **Affinity Diagramming, Multi-Criteria Evaluation, Sketching, and Dot Voting**
- Provides **ready-to-prototype concepts for MUP design**

Period 3

Minimum Usable Prototype (MUP) Design

"MUP Design = Creating a simple, functional version of a solution to test its value, usability, and feasibility with real users."

1.What is a Minimum Usable Prototype

Definition:

A **Minimum Usable Prototype (MUP)** is a **simplified version of a solution concept that demonstrates core functionalities or benefits** to users.

Key Points:

- Focuses on **core features only**
- Allows **quick testing and feedback**
- Reduces **time, cost, and risk**

- Supports **iterative learning and improvement**

2.Objectives of MUP Design

1. **Validate Core Value:**
   - ✓ Test if the **solution provides real value to users**
2. **Test Usability:**
   - ✓ Assess if users can **easily use the solution**
3. **Check Feasibility:**
   - ✓ Determine **technical or practical limitations** early
4. **Support Iteration:**
   - ✓ Gather **feedback to refine the concept**

3.Steps in MUP Design

1. **Identify Core Features:**
   - ✓ Focus on **key functionality** that delivers value
2. **Select Prototype Type:**
   - ✓ Paper prototype, digital mockup, clickable app, or role-play
3. **Build the Prototype:**
   - ✓ Keep it **simple, low-cost, and quick to develop**
4. **Define Test Scenarios:**
   - ✓ Prepare tasks for users to perform with the MUP
5. **Gather Feedback:**
   - ✓ Observe user interactions, take notes, and ask questions
6. **Refine and Iterate:**
   - ✓ Improve the prototype based on **feedback and learning**

4. Techniques for MUP Design

Technique	Description	Example
Paper Prototyping	Sketch solution interfaces on paper	Draw screens of classroom booking app
Clickable Wireframes	Low-fidelity digital mockups with basic	Use Figma or Adobe XD to simulate app flow

Technique	Description	Example
	interactivity	
Role-Playing / Wizard-of-Oz	Simulate solution manually without full system	Teacher acts as app backend providing real-time room info
Storyboards	Visual storytelling of user interaction	Illustrate a student using the app from search to booking
Rapid Iteration	Quick cycles of prototype, test, refine	Test initial app screens and adjust layout or features

5.Example – Engineering Scenario

Problem

Maintenance engineers struggle to quickly identify faults in industrial machines, leading to delayed repairs and production losses.

Selected Concept

Sensor-based machine monitoring system + diagnostic dashboard

MUP (Minimum Usable Product) Design:

1. **Core Features:**
   - ✓ Live machine status (temperature, vibration, pressure)
   - ✓ Fault alerts with severity levels
   - ✓ Simple dashboard showing healthy vs faulty machines
2. **Prototype Type:**
   - ✓ Low-fidelity dashboard mock-up (Figma)
   - ✓ Simulated sensor data displayed on screen
3. **Test Scenario:**
   - ✓ Engineers monitor machines and identify a fault using the MUP
   - ✓ Take action based on alert information

4. **Feedback Collected:**
    - ✓ Clarity of fault alerts
    - ✓ Ease of interpreting sensor data
    - ✓ Usefulness for decision-making
5. **Iteration:**
    - ✓ Add color coding for fault severity
    - ✓ Simplify graphs into indicators
    - ✓ Include recommended maintenance actions

Outcome

The MUP validated that real-time monitoring improves fault identification and supports faster maintenance decisions before full system deployment.

6.Classroom Activity

Activity: Build a Simple MUP

Steps:

1. Form small groups
2. Choose **one shortlisted concept from Period 2**
3. Identify **core features to test**
4. Create a **paper or digital MUP**
5. Conduct a **mini-test with peers**
6. Note observations and suggest **improvements for iteration**

Purpose:

- Experience **rapid prototyping and testing**
- Reinforce **focus on core value and usability**
- Prepare for **concept evaluation and refinement in upcoming periods**

7.Best Practices

- Focus on **core functionality, not full features**
- Use **low-cost materials** for rapid iteration
- Test early with **representative users**
- Observe **user behavior and feedback carefully**
- Iterate quickly to improve the prototype

8.Common Mistakes

☐ Trying to build a full-featured solution too early
☐ Ignoring user feedback or observations
☐ Overcomplicating the prototype, making testing difficult
☐ Skipping iteration cycles

9.Key Takeaways

- MUP Design is about **creating a simple, testable version of a concept**
- Focuses on **core features, value, usability, and feasibility**
- Enables **rapid feedback and iterative improvement**
- Reduces **risk, cost, and time** before full development

Period 4

Conceptualize the Solution Concept

"Conceptualizing the Solution Concept = Creating a structured and well-defined blueprint of the solution that clearly communicates what it does, who it serves, and how it delivers value."

1.What is Conceptualizing a Solution Concept?

Definition:

Conceptualizing a solution concept is the process of **defining the solution clearly in terms of user needs, core functionality, benefits, and implementation approach**.

Key Points:

- Transforms **ideas and MUPs into a coherent solution concept**
- Focuses on **user value, usability, feasibility, and uniqueness**
- Serves as a **foundation for prototyping and testing**
- Ensures **clarity among team members and stakeholders**

2.Steps to Conceptualize a Solution Concept

1. **Define the Problem Clearly:**
   - ✓ Restate the validated problem using **user-centered language**
2. **Identify the Target User / Persona:**
   - ✓ Specify **who will use the solution and their needs**
3. **Determine Core Features / Functions:**
   - ✓ Focus on **what is essential to deliver value**
4. **Describe Benefits and Value Proposition:**
   - ✓ Explain **how the solution solves the problem and why it matters**
5. **Consider Feasibility and Constraints:**
   - ✓ Assess **technical, financial, and practical feasibility**
6. **Visualize the Concept:**
   - ✓ Use **sketches, diagrams, or storyboards** to communicate the solution

7. **Iterate and Refine:**
    - ✓ Update the concept based on **team discussions and user feedback**

3.Why Conceptualizing is Important

1. **Provides Clarity:**
    - ✓ Everyone understands **what the solution is and how it works**
2. **Guides Prototype Development:**
    - ✓ Serves as a **blueprint for building MUPs**
3. **Supports Evaluation:**
    - ✓ Helps assess **usability, feasibility, and value** before full development
4. **Reduces Risk:**
    - ✓ Prevents **misunderstandings and wasted effort**

4.Techniques for Conceptualization

Technique	Purpose	Example
Concept Statement	Summarizes problem, solution, target users, and benefits	"A mobile app for students to find empty classrooms in real-time, saving time and reducing stress"
Sketching / Diagramming	Visualize solution flow and core features	Draw app screens showing search, map view, and notifications
Storyboarding	Show how users interact with the solution step by step	Illustrate a student searching and booking a classroom
Feature Prioritization	Identify essential vs. optional features	Highlight free classroom display and notification as core, booking as optional
Team Discussion & Iteration	Refine concept based on feedback	Brainstorm improvements for usability and engagement

5. Example – Engineering Scenario

Problem

Manufacturing engineers find it difficult to monitor energy consumption of machines, leading to inefficient power usage and higher operational costs.

Conceptualized Solution:

- **Target User:**
  Manufacturing and plant engineers
- **Core Features:**
  - ✓ Real-time energy consumption monitoring
  - ✓ Machine-wise energy usage dashboard
  - ✓ Alerts for abnormal power consumption
- **Benefits / Value Proposition:**
  Improves energy efficiency, reduces power costs, and supports data-driven decision-making
- **Feasibility:**
  Uses existing smart energy meters, IoT sensors, and a web-based monitoring system
- **Visualization:**
  Sketch MUP dashboard screens showing live energy graphs, alerts, and machine status

Outcome

A clear and structured engineering solution that is ready for prototyping and on-site testing in an industrial environment.

6. Classroom Activity

Activity: Conceptualize a Solution Concept

Steps:

1. Form small groups
2. Select a **validated problem and persona**
3. Define the **problem, target user, core features, and benefits**
4. Visualize the solution using **sketches or storyboards**
5. Share with class and gather **peer feedback for refinement**

Purpose:

- Practice **structuring ideas into a coherent solution concept**
- Reinforce **user-centered thinking**
- Prepare for **prototype development and evaluation**

7. Best Practices

- Focus on **core value and essential features**
- Keep the **user perspective central**
- Use **visualization to communicate clearly**
- Iterate based on **team and user feedback**

8.Common Mistakes

☐ Skipping visualization and documentation
☐ Including unnecessary features that complicate the concept
☐ Ignoring user needs or feasibility constraints
☐ Failing to iterate based on feedback

9. Key Takeaways

- Conceptualization **transforms ideas and MUPs into structured, actionable solution concepts**

- Focus on **user, core features, benefits, and feasibility**
- Uses **sketches, storyboards, and concept statements**
- Forms the **foundation for prototyping, testing, and refinement**

Period 5

Explore, Iterate, and Learn

"Explore, Iterate, and Learn = Test ideas and prototypes, learn from feedback, and refine solutions continuously."

1. What is Explore, Iterate, and Learn?

Definition:

Explore, Iterate, and Learn is an **iterative approach to refining solution concepts and prototypes** based on real-world feedback, experimentation, and observation.

Key Points:

- Focuses on **learning through action and experimentation**
- Encourages **rapid iterations** to improve the solution
- Ensures solutions **meet user needs effectively**
- Reduces **risks of building the wrong product**

2.Steps in the Explore, Iterate, and Learn Process

1. **Explore:**
    - ✓ Test prototypes or concepts in **realistic scenarios**
    - ✓ Collect **qualitative and quantitative feedback** from users

2. **Observe and Document:**
    - ✓ Note **user interactions, challenges, and preferences**
    - ✓ Identify **pain points and areas for improvement**
3. **Analyze Feedback:**
    - ✓ Compare results against **user needs and value proposition**
    - ✓ Identify patterns and recurring issues
4. **Iterate:**
    - ✓ Make **small, focused improvements** to the concept or prototype
    - ✓ Prioritize changes that **maximize value and usability**
5. **Learn and Refine:**
    - ✓ Update the solution based on **feedback and insights**
    - ✓ Repeat cycles until the **solution meets core user needs and usability goals**

3.Why This Process is Important

1. **Reduces Risk:**
    - ✓ Prevents **investing in untested or unusable solutions**
2. **Encourages Learning:**
    - ✓ Helps teams **understand user behavior and preferences**
3. **Improves Solution Quality:**
    - ✓ Iterative improvements **enhance usability, functionality, and value**
4. **Supports User-Centric Design:**
    - ✓ Keeps **real user needs at the center** of the development process

4.Techniques for Exploring, Iterating, and Learning

Technique	Description	Example
User Testing	Observe users interacting with the MUP	Students try the classroom app to find available rooms
A/B Testing	Compare two versions of the prototype	Test two notification styles for clarity and appeal
Feedback Sessions	Collect structured user input	Ask students what they like/dislike about the prototype
Rapid Iteration	Make small changes quickly based on feedback	Adjust map view, notification colors, or icons
Analytics & Observation	Record how users perform tasks	Measure time taken to find a classroom using MUP

5.Example

Problem:

Structural engineers make design errors due to misinterpretation of load data and unclear visualization of stress zones.

Process:

1. **Explore:**
   Engineers use an initial MUP that displays structural load calculations and simulation outputs in numerical tables.
2. **Observe:**
   Users find it difficult to interpret critical stress areas because values are abstract and lack visual context.
3. **Analyze Feedback:**
   Engineers request graphical stress maps, clearer legends, and warnings when safety limits are exceeded.
4. **Iterate:**
   The MUP is updated with color-graded stress visualizations, safety threshold indicators, and simplified legends.

5. **Learn:**
   Engineers better understand structural behavior and identify risky designs earlier.

Outcome:
Iterative cycles improved design accuracy, safety compliance, and engineer confidence.

6. Classroom Activity

Activity: Explore, Iterate, and Learn

Steps:

1. Form small groups
2. Use your **MUP from previous periods**
3. Conduct a **mini user test with peers**
4. Observe and document issues or suggestions
5. Make **quick iterative changes** to improve the prototype
6. Share **learning outcomes and improvements** with the class

Purpose:

- Practice **iterative prototyping and feedback incorporation**
- Reinforce **user-centered, experimental learning**
- Prepare solutions for **evaluation of usability, capability, and feasibility**

7.Best Practices

- Test **early and often**, even with simple prototypes
- Focus on **learning, not perfection** in the first iteration
- Collect **objective observations, not assumptions**
- Prioritize changes that **maximize user value and usability**

8.Common Mistakes

 Skipping feedback collection and iteration

☐ Making large changes without testing

☐ Ignoring user observations in favor of personal assumptions

☐ Failing to document learning for the next cycle

9.Key Takeaways

- **Explore, Iterate, and Learn** is the backbone of **user-centered design**
- Rapid cycles of testing and improvement **reduce risk and improve usability**
- Techniques include **user testing, feedback sessions, rapid iteration, and observation**
- Iteration ensures solutions **deliver real value and meet user needs effectively**

Period 6

Build the Right Prototype

"Build the Right Prototype = Focused, efficient prototyping that tests core functionalities and delivers maximum learning with minimal effort."

1.What Does "Build the Right Prototype" Mean?

Definition:

Building the right prototype is the process of **designing a prototype that effectively tests critical features, validates**

assumptions, and delivers insights, without overbuilding unnecessary aspects.

Key Points:

- Focuses on **core value propositions**
- Avoids **overcomplicating the prototype**
- Ensures **user feedback is actionable**
- Saves **time, cost, and resources**

2.Steps to Build the Right Prototype

1. **Identify Core Value and Features:**
   - ✓ Determine the **essential features** needed to validate the concept
2. **Select Prototype Type:**
   - ✓ Paper prototype, digital mockup, clickable wireframe, or role-play
3. **Prioritize Testing Goals:**
   - ✓ Define **what you want to learn** from users (usability, desirability, feasibility)
4. **Design Efficiently:**
   - ✓ Focus on **simplicity, clarity, and speed of development**
5. **Test with Users:**
   - ✓ Observe interactions, gather feedback, and collect metrics
6. **Iterate:**
   - ✓ Refine prototype based on insights from **real user testing**

3.Why Building the Right Prototype is Important

1. **Maximizes Learning:**
   - ✓ Focused prototypes help **quickly validate critical assumptions**
2. **Reduces Waste:**
   - ✓ Avoids building **features that do not contribute to value**
3. **Supports User-Centered Design:**
   - ✓ Ensures the prototype **aligns with user needs**
4. **Enables Iteration:**
   - ✓ Simplified prototypes are **easier and faster to improve**

4.Techniques to Build the Right Prototype

Technique	Purpose	Example
Paper Prototyping	Simple, fast sketches for testing flow	Draw app screens showing classroom availability
Clickable Wireframes	Low-fidelity digital mockup for basic interactions	Figma prototype for mobile app navigation
Role-Playing / Wizard-of-Oz	Simulate backend manually	Teacher provides real-time updates as the app
Feature Prioritization	Focus on essential vs optional features	Show free classrooms first, booking feature optional
Scenario-Based Testing	Test prototypes in real use cases	Students find a classroom using the MUP under realistic conditions

5. Example

Problem:

Workers are exposed to safety risks because hazard warnings are unclear or easily overlooked.

Build the Right Prototype:

1. **Core Features:**

   Real-time hazard alerts, safety zone indicators, and emergency instructions.

2. **Prototype Type:**
   Paper prototype of warning displays + interactive mobile/helmet HUD mockup.
3. **Testing Goal:**
   Determine whether workers can quickly recognize hazards and understand required safety actions.
4. **Iterate:**
   Add high-contrast color codes, universal safety icons, vibration alerts, and simplified messages based on feedback.
5. **Outcome:**
   The prototype validates improved hazard awareness and safety compliance without adding unnecessary features.

6. Classroom Activity

Activity: Build the Right Prototype

Steps:

1. Form small groups
2. Use your **concept from previous periods**
3. Identify **essential features for testing**
4. Build a **prototype (paper or digital)** focusing only on core value
5. Conduct **peer testing** and collect feedback
6. Note improvements and plan **next iteration**

Purpose:

- Reinforces **focus on core features and learning goals**
- Encourages **efficient prototyping and iteration**
- Prepares for **evaluation of usability, capability, and feasibility in upcoming periods**

7.Best Practices

- Focus on **testing assumptions, not full features**
- Keep prototypes **simple, low-cost, and fast to build**
- Observe users carefully and **document insights**
- Iterate quickly based on **feedback**
- Prioritize **features that deliver the most user value**

8.Common Mistakes

☐ Trying to build a fully functional product too early
☐ Including unnecessary features in the prototype
☐ Ignoring testing goals or learning objectives
☐ Skipping user testing before iteration

9.Key Takeaways

- Building the right prototype ensures **focused testing and learning**
- Core features must **align with user value and usability**
- Use **paper prototypes, wireframes, role-play, or MUPs** efficiently
- Supports **iterative design, rapid learning, and risk reduction**

Period 7

Assess Capability

"Assess Capability = Evaluate whether the prototype or concept can successfully perform its intended functions and meet user needs."

1. What is Capability Assessment?

Definition:

Capability assessment is the process of **evaluating a prototype or concept to determine if it can deliver the desired functionality, meet requirements, and satisfy user needs**.

Key Points:

- Focuses on **technical, functional, and operational aspects**
- Identifies **strengths, limitations, and improvement areas**
- Ensures **the solution can actually work in real-world conditions**

2.Steps to Assess Capability

1. **Define Key Capabilities to Test:**
   - ✓ Identify **essential functions and features** the solution must perform
2. **Set Evaluation Criteria:**
   - ✓ Performance, reliability, accuracy, responsiveness, and consistency
3. **Test the Prototype or Concept:**
   - ✓ Conduct simulations, role-playing, or technical testing
4. **Document Observations:**
   - ✓ Note which features **work well, which need improvement**
5. **Analyze Gaps and Limitations:**
   - ✓ Identify where capability is insufficient or suboptimal
6. **Plan Improvements:**
   - ✓ Suggest modifications or enhancements for the next iteration

3.Why Assessing Capability is Important

1. **Validates Functionality:**
   - ✓ Ensures the solution **can perform its intended tasks**
2. **Identifies Limitations Early:**
   - ✓ Reduces **risk of failure during full-scale implementation**
3. **Supports Iterative Improvement:**
   - ✓ Provides **feedback for refining the solution**
4. **Aligns Solution with User Needs:**
   - ✓ Confirms the prototype **meets expectations and requirements**

4.Techniques for Capability Assessment

Technique	Purpose	Example
Task Testing	Observe users performing core tasks	Students use the app to find and book classrooms
Simulation / Role-Play	Test solution under controlled scenarios	Teacher acts as backend system providing updates
Checklist Evaluation	Compare features against required capabilities	Check if notifications, map, and search features work
Performance Metrics	Measure speed, accuracy, or efficiency	Record time taken to locate an available classroom
Peer Review	Get feedback from team members	Discuss functional gaps and possible improvements

5. Example

Problem:

Electronics engineers struggle to detect and diagnose circuit faults due to unclear signal monitoring and alerts.

Prototype:

Circuit monitoring system + digital diagnostic dashboard

Capability Assessment:

1. **Core Features to Test:**
   Real-time voltage and current monitoring, fault alerts, signal waveform visualization
2. **Evaluation Criteria:**
   Accuracy of signal readings, clarity of fault alerts, readability of waveforms
3. **Testing:**
   Engineers analyze a test circuit using the MUP under normal and fault conditions
4. **Observations:**
   Waveforms are clear, but fault notifications lack sufficient context for diagnosis
5. **Improvements:**
   Add fault location indicators, color-coded signal thresholds, and concise diagnostic hints

Outcome:
The prototype demonstrates reliable monitoring and fault detection, requiring minor refinements to improve diagnostic clarity and usability.

6.Classroom Activity

Activity: Assess Capability of Prototype

Steps:

1. Form small groups
2. Use your **prototype from previous periods**
3. Define **key capabilities to test**
4. Conduct **peer testing or role-play**
5. Document **observations and gaps**

6. Suggest **improvements for iteration**

Purpose:

- Reinforces **objective evaluation of functionality**
- Highlights **strengths and areas for improvement**
- Prepares prototype for **usability and feasibility assessment in upcoming periods**

7.Best Practices

- Focus on **core features first**
- Use **realistic scenarios** for testing
- Document results systematically for **future iterations**
- Engage **multiple users or team members** to validate findings

8.Common Mistakes

 Ignoring key capabilities or testing only superficially
☐ Failing to define clear evaluation criteria
☐ Skipping documentation of gaps and improvements
☐ Assuming a working prototype is automatically fully capable

9.Key Takeaways

- Assessing capability ensures the **solution can perform intended functions effectively**
- Focus on **core features, performance, and operational feasibility**
- Techniques include **task testing, simulation, checklists, performance metrics, and peer review**
- Provides **critical insights for refinement before usability and feasibility testing**

Period 8

Assess Usability and Feasibility

"Assess Usability and Feasibility = Evaluate whether the solution is easy to use, satisfies user needs, and is practical to implement."

1.What is Usability and Feasibility Assessment?

Definitions:

Usability Assessment:

- Measures how **easy and efficient it is for users to interact with the prototype or solution**.
- Focuses on **intuitiveness, learnability, and user satisfaction**.

Feasibility Assessment:

- Determines whether the solution is **practical to implement given technical, financial, operational, and resource constraints**.

Key Points:

- Both assessments are **critical before full-scale development**
- Usability ensures **user adoption and satisfaction**
- Feasibility ensures **solution can be implemented successfully**

2.Steps to Assess Usability

1. **Define User Tasks:**
   - ✓ Identify key tasks users need to perform
2. **Conduct Usability Testing:**
   - ✓ Observe users performing tasks using the prototype

3. **Collect Feedback:**
    - ✓ Ask users about **ease of use, satisfaction, and clarity**
4. **Identify Usability Issues:**
    - ✓ Record challenges, confusion, or errors
5. **Refine Prototype:**
    - ✓ Improve design, layout, navigation, and interactions

3.Steps to Assess Feasibility

1. **Define Implementation Requirements:**
    - ✓ Technical, financial, operational, and resource requirements
2. **Analyze Technical Feasibility:**
    - ✓ Can the solution be built with existing technology?
3. **Analyze Financial Feasibility:**
    - ✓ Is the solution cost-effective and sustainable?
4. **Analyze Operational Feasibility:**
    - ✓ Can it be implemented within existing workflows or processes?
5. **Document Feasibility Findings:**
    - ✓ Identify potential challenges and mitigation strategies

4.Techniques for Usability and Feasibility Assessment

Technique	Purpose	Example
Usability Testing	Evaluate ease of use and efficiency	Students test the classroom app for finding and booking rooms
Heuristic Evaluation	Check design against usability principles	Evaluate map clarity, notification visibility, and button placement
Surveys & Feedback Forms	Collect user satisfaction ratings	Ask students to rate navigation, clarity, and usefulness
Feasibility Analysis	Evaluate technical, financial, and operational viability	Check app compatibility with devices, server requirements, and budget

Technique	Purpose	Example
		constraints
Pilot Implementation	Test solution in a small real-world scenario	Deploy the app for one building or one floor on campus

5.Example

Problem:

Machine operators experience difficulty detecting abnormal operating conditions in industrial machinery.

Prototype:

Condition-monitoring system with vibration sensors and a machine health dashboard

Usability Assessment:

- Operators perform tasks: monitor machine status, check vibration levels, receive alerts
- **Observations:** Dashboard is generally clear, but some graphs and symbols are hard to interpret
- **Iteration:** Simplify graphs, add color-coded condition indicators, improve alert messages

Feasibility Assessment:

- **Technical:** Works with existing vibration sensors, PLCs, and industrial networks
- **Financial:** Sensor installation and software development costs fit within maintenance budget
- **Operational:** Can be integrated into current preventive maintenance workflows

Outcome:

The prototype is usable, feasible, and suitable for final refinement before full-scale deployment.

6.Classroom Activity

Activity: Assess Usability and Feasibility

Steps:

1. Form small groups
2. Use your **prototype from previous periods**
3. Conduct **usability testing with peers**
4. Collect **feedback on ease of use, efficiency, and clarity**
5. Evaluate **feasibility** in terms of technology, cost, and implementation
6. Document findings and propose **improvements**

Purpose:

- Reinforces **user-centered evaluation**
- Ensures solution is both **practical and usable**
- Prepares solution for **final refinement and presentation**

7.Best Practices

- Test with **representative users** for accurate insights
- Focus on **critical tasks first**
- Keep feasibility assessment **realistic and evidence-based**
- Document all findings for **iteration and decision-making**

8.Common Mistakes

 Skipping usability testing and assuming the prototype is intuitive

☐ Ignoring technical or financial constraints

☐ Failing to document findings for iteration

☐ Making assumptions about user satisfaction without testing

9.Key Takeaways

- Usability ensures the solution is **easy to use, intuitive, and user-friendly**
- Feasibility ensures the solution is **practical, sustainable, and implementable**
- Both assessments are **critical before scaling or finalizing the solution**
- Techniques include **usability testing, heuristic evaluation, surveys, feasibility analysis, and pilot implementation**

Period 9

Systematic Concept Generation & Evaluation of Technology Alternatives

"Systematic Concept Generation = Structured approach to developing solution concepts.
Evaluation of Technology Alternatives = Assessing different technologies to implement the solution effectively."

1.What is Systematic Concept Generation?

Definition:

Systematic concept generation is a **structured method to develop and refine multiple solution concepts** based on validated user needs, feasibility, and value.

Key Points:

- Moves beyond random brainstorming
- Encourages **structured exploration of ideas**
- Focuses on **user needs, constraints, and feasibility**
- Prepares multiple solution concepts for **comparison and selection**

2.Steps in Systematic Concept Generation

1. **Define Problem Clearly:**
   - ✓ Use validated problem statements from research and previous iterations
2. **Set Evaluation Criteria:**
   - ✓ Features, usability, value, feasibility, cost, technical constraints
3. **Generate Concepts:**
   - ✓ Use structured techniques (e.g., SCAMPER, morphological analysis, or TRIZ)
4. **Document Each Concept:**
   - ✓ Include user needs addressed, core features, benefits, and constraints
5. **Screen Concepts:**
   - ✓ Eliminate low-value or infeasible concepts
6. **Refine Concepts:**
   - ✓ Combine complementary ideas to strengthen solutions

3.Evaluation of Technology Alternatives

Definition:

Evaluation of technology alternatives is the process of analyzing different technological options for implementing the solution concept, ensuring optimal performance, cost-effectiveness, and feasibility.

Key Points:

- Considers **hardware, software, platforms, and tools**
- Evaluates **cost, scalability, reliability, and integration**
- Reduces **risk of selecting inappropriate technology**

4. Steps to Evaluate Technology Alternatives

1. **Identify Requirements:**
   - ✓ Define what the technology must support (e.g., platform compatibility, performance, user load)
2. **List Alternatives:**
   - ✓ Research multiple options for software, hardware, or platforms
3. **Define Evaluation Criteria:**
   - ✓ Cost, scalability, maintainability, integration, reliability
4. **Compare Alternatives:**
   - ✓ Use scoring, pros/cons, or weighted decision matrix
5. **Select Optimal Technology:**
   - ✓ Choose technology that **best supports the solution concept and project constraints**

5.Techniques for Systematic Concept Generation & Technology Evaluation

Technique	Purpose	Example
Morphological Analysis	Explore multiple solutions systematically	Combine different app features (map, notifications, booking) in structured matrix
TRIZ (Theory of Inventive Problem Solving)	Generate innovative solutions	Use inventive principles to overcome classroom booking constraints
Weighted Decision Matrix	Compare technology options objectively	Score mobile app platforms (Android, iOS, web) based on cost, scalability, and ease of development
Prototyping with Alternatives	Test different technologies in small	Build a paper prototype vs digital mockup to see which works best

Technique	Purpose	Example
	MUPs	
Scenario Analysis	Evaluate solution in real-world context	Simulate app usage during peak classroom hours to test performance

6.Example

Problem:

Machines overheat without clear indication to operators.

Systematic Concept Generation:

- **Concept 1:** Temperature gauge mounted on the machine
- **Concept 2:** Warning light that turns on when temperature exceeds limit
- **Concept 3:** Mobile alert system connected to temperature sensor

Evaluation of Technology Alternatives:

- **Temperature Measurement:** Analog sensor vs digital temperature sensor
- **Alert Method:** Warning light vs buzzer vs mobile notification
- **Display:** Analog dial vs simple digital display

Outcome:

- **Selected Concept:** Digital temperature sensor with warning light
- **Technology:** Low-cost sensor and LED indicator, easy to install and maintain

7. Classroom Activity

Activity: Systematic Concept Generation & Technology Evaluation

Steps:

1. Form small groups

2. Generate **3–5 solution concepts systematically** using SCAMPER or Morphological Analysis
3. List **technological options** for each concept
4. Evaluate alternatives using **weighted decision matrix**
5. Select **best solution concept and technology** for final prototype development

Purpose:

- Reinforces **structured concept generation**
- Encourages **critical thinking in technology selection**
- Prepares students for **effective and practical solution implementation**

8.Best Practices

- Use **structured frameworks** to reduce bias in concept generation
- Evaluate **all feasible technological alternatives**
- Include **cost, scalability, and user needs** in decision-making
- Document **selection rationale** for clarity and future reference

9.Key Takeaways

- Systematic concept generation ensures **structured and high-quality solution concepts**
- Technology evaluation ensures the solution is **practical, scalable, and cost-effective**
- Techniques like **Morphological Analysis, TRIZ, and Weighted Decision Matrix** help in **objective selection**
- Combining both ensures a **ready-to-develop, user-centered, and technically viable solution**

UNIT V

SYSTEM THINKING

Period 1

System Thinking

"System Thinking = Understanding the whole system, including relationships, feedback loops, and patterns, rather than just individual parts."

1.What is System Thinking?

Definition:

System Thinking is a **holistic approach** to analysis that focuses on **how parts of a system interrelate and how systems work over time within the context of larger systems**.

Key Points:

- Focuses on **relationships, interdependencies, and feedback loops**
- Helps identify **root causes of problems** rather than just symptoms
- Encourages **long-term thinking and sustainable solutions**

2.Key Concepts in System Thinking

1. **System:**
   - ✓ A set of elements interacting to form a whole
   - ✓ Example: Campus ecosystem – students, classrooms, schedules, staff
2. **Components / Elements:**
   - ✓ Individual parts of the system
   - ✓ Example: Classroom availability, mobile app, Wi-Fi network

3. **Interconnections:**
    - ✓ Relationships between components
    - ✓ Example: Classroom occupancy affects student movement, notifications, and app usage
4. **Purpose / Function:**
    - ✓ The intended outcome or goal of the system
    - ✓ Example: Efficient use of classrooms and improved student productivity
5. **Feedback Loops:**
    - ✓ Circular cause-and-effect relationships
    - ✓ Positive feedback: Reinforces system behavior
    - ✓ Negative feedback: Balances or stabilizes the system

3. Why System Thinking is Important

1. **Understand Complexity:**
    - ✓ Helps see how multiple factors influence each other
2. **Solve Problems Holistically:**
    - ✓ Identifies root causes instead of treating symptoms
3. **Predict Consequences:**
    - ✓ Understand how changes in one part affect the entire system
4. **Support Innovation:**
    - ✓ Helps design **solutions that work effectively in complex environments**

4.Examples of Systems

System Type	Example	Explanation
Natural System	Ecosystem, weather	Elements interact (plants, animals, climate) to maintain balance
Human-Made System	Traffic management, public transport	Roads, vehicles, signals interact to influence flow

System Type	Example	Explanation
Organizational System	Campus scheduling system	Classrooms, students, professors, and notifications interact to manage resources
Complex Adaptive System	Social networks, e-commerce platforms	Many interacting components that adapt over time to changes

---

5.Steps to Apply System Thinking

1. **Identify the System:**
   - ✓ Define boundaries and components
2. **Analyze Interconnections:**
   - ✓ Understand relationships, dependencies, and feedback loops
3. **Identify Patterns and Trends:**
   - ✓ Observe recurring behaviors over time
4. **Understand System Behavior:**
   - ✓ Analyze how changes in one part affect the whole system
5. **Develop Solutions Holistically:**
   - ✓ Design interventions that **consider the entire system**

6.Classroom Activity

Activity: Map a Simple System

Steps:

1. Form small groups
2. Select a simple system (e.g., campus classroom allocation)
3. Identify **components and interconnections**
4. Draw a **system map showing elements, relationships, and feedback loops**
5. Discuss how changes in one component affect the system

Purpose:

- Introduces students to **visualizing systems**

- Helps understand **interdependencies and complexity**
- Prepares for analyzing **complex systems in later periods**

7.Best Practices

- Look at **the system as a whole, not just individual parts**
- Identify **feedback loops and interconnections**
- Consider **long-term effects of changes**
- Use **visual tools like system maps, causal loop diagrams**

8.Common Mistakes

☐ Focusing only on individual components
☐ Ignoring feedback loops and relationships
☐ Treating symptoms rather than root causes
☐ Oversimplifying complex systems

9.Key Takeaways

- System Thinking helps **understand complex systems holistically**
- Focus on **components, interconnections, purpose, and feedback loops**
- Useful in **innovation, problem-solving, and strategic planning**
- Tools like **system maps and causal diagrams** aid understanding

---

Period 2

Understanding Systems

"Understanding Systems = Analyzing system components, relationships, and behaviors to comprehend how the system functions as a whole."

1. What Does "Understanding Systems" Mean?

Definition:

Understanding systems is the process of **analyzing a system's components, their interconnections, behavior patterns, and the purpose of the system** to predict outcomes and design better solutions.

Key Points:

- Helps identify **root causes of problems**
- Reveals **interdependencies and potential ripple effects**
- Encourages **long-term thinking and holistic solutions**
- Supports **innovation and decision-making** in complex environments

2.Key Elements in Understanding Systems

1. **Components / Elements:**
   - ✓ Parts of the system that perform specific functions
   - ✓ Example: Students, classrooms, faculty, schedules
2. **Interconnections / Relationships:**
   - ✓ How components influence each other
   - ✓ Example: Classroom allocation affects student movement and notifications

3. **Feedback Loops:**
    - ✓ Circular cause-and-effect that shapes system behavior
    - ✓ **Positive feedback:** Amplifies change (e.g., more students using app → faster adoption)
    - ✓ **Negative feedback:** Stabilizes system (e.g., scheduling limits → prevents overcrowding)
4. **System Boundaries:**
    - ✓ Defines what is **inside and outside** the system
    - ✓ Example: University campus system boundary includes classrooms, students, staff but excludes other campuses
5. **Purpose / Goal:**
    - ✓ The intended outcome of the system
    - ✓ Example: Efficient classroom utilization and improved student productivity

3.Types of System Behavior

Behavior Type	Description	Example
Linear Behavior	Output changes proportionally with input	Adding one classroom increases capacity by one
Non-linear Behavior	Small input changes cause large effects	Adding more students leads to congestion beyond a threshold
Emergent Behavior	New patterns emerge from interactions	Students adapt their schedules based on notifications and room availability
Dynamic Behavior	Changes over time with feedback loops	Classroom occupancy fluctuates daily based on courses and student habits

4.Techniques to Understand Systems

1. **System Mapping:**
    - ✓ Visualize components and their interactions

- ✓ Example: Map classrooms, students, staff, and scheduling interactions

2. **Causal Loop Diagrams:**
   - ✓ Illustrate feedback loops and cause-effect relationships
   - ✓ Example: More app notifications → more classroom usage → improved efficiency
3. **Stock and Flow Analysis:**
   - ✓ Understand accumulation and movement within the system
   - ✓ Example: Number of available classrooms (stock) changes with course schedules (flow)
4. **Scenario Analysis:**
   - ✓ Explore how changes affect system behavior
   - ✓ Example: What happens if more students enroll without adding classrooms?

5.Example – Air Conditioning System in a Building

System: Building temperature control system

Components: Temperature sensors, air conditioner, rooms, occupants, control unit

Interconnections:

- Room temperature affects sensor readings
- Sensor readings control AC operation

Feedback Loops:

- High temperature → AC turns on → temperature decreases → AC turns off

Behavior Observed:

Emergent behavior – stable and comfortable indoor temperature is maintained automatically

Outcome:

Understanding the system helps engineers design energy-efficient temperature control systems.

6. Classroom Activity

Activity: Analyze a System

Steps:

1. Form small groups
2. Choose a **simple system from campus or daily life**
3. Identify **components, interconnections, boundaries, and purpose**
4. Draw a **system map or causal loop diagram**
5. Discuss how **changes in one component affect the system**

Purpose:

- Reinforces **system thinking and analysis**
- Helps students **visualize and understand complex interactions**
- Prepares for **studying examples of complex systems in upcoming periods**

7.Best Practices

- Identify **all relevant components and relationships**
- Consider **feedback loops and dynamic behavior**
- Define **system boundaries clearly**
- Focus on **purpose and goals of the system**

- Use **visual tools to represent complexity**

8. Common Mistakes

 Focusing on individual components instead of the whole system
☐ Ignoring feedback loops and interdependencies
☐ Overlooking system boundaries
☐ Assuming linear behavior in inherently complex systems

9.Key Takeaways

- Understanding systems requires **analyzing components, interconnections, boundaries, and goals**
- Feedback loops and emergent behaviors are critical to predict outcomes
- Tools like **system maps, causal loop diagrams, and scenario analysis** aid comprehension
- Helps in **designing holistic, sustainable solutions in complex environments**

Period 3

Examples of Systems

"*Examples of Systems = Real-world systems that demonstrate interconnections, feedback loops, and complexity.*"

1.Importance of Studying System Examples

- Makes **abstract concepts tangible**
- Shows **how systems behave in real life**
- Helps identify **patterns, feedback loops, and emergent behavior**
- Encourages **holistic problem-solving**

- Prepares students to **apply system thinking in design and innovation**

2.Categories of Systems

1. **Natural Systems**
   - ✓ Systems that occur in nature, with interacting components
   - ✓ **Example:** Ecosystem – plants, animals, climate, and soil interact to maintain balance
2. **Human-Made Systems**
   - ✓ Designed systems by humans to serve specific purposes
   - ✓ **Example:** Traffic management system – roads, vehicles, signals, and rules interact to manage flow
3. **Organizational Systems**
   - ✓ Systems within institutions or organizations
   - ✓ **Example:** University campus system – students, classrooms, faculty, scheduling, notifications
4. **Complex Adaptive Systems**
   - ✓ Systems with multiple interacting components that **adapt and evolve**
   - ✓ **Example:** Social networks – user behavior affects connections and information flow
   - ✓ **Example:** E-commerce platforms – buyer and seller interactions, algorithms, and promotions

3.Detailed Examples

System Type	Example	Components	Interconnections	Observations / Learning
Natural	Forest ecosystem	Trees, animals, soil, water, climate	Predator-prey relationships, nutrient cycles	Shows **interdependence and feedback loops**
Human-Made	Traffic system	Roads, vehicles, traffic signals, pedestrians	Traffic flow depends on signal timing, vehicle speed	Demonstrates **linear and non-linear**

System Type	Example	Components	Interconnections	Observations / Learning
				behavior
Organizational	Campus classroom allocation	Classrooms, students, faculty, scheduling system, notifications	Schedule affects student movement and classroom usage	Example of **dynamic behavior and emergent patterns**
Complex Adaptive	Online social network	Users, posts, interactions, algorithms	Interactions influence trends and engagement	Demonstrates **emergent behavior and adaptability**

4.How to Analyze Examples

1. **Identify Components:**
   - ✓ What are the key elements?
2. **Understand Interconnections:**
   - ✓ How do elements influence one another?
3. **Observe Feedback Loops:**
   - ✓ Are there reinforcing or balancing loops?
4. **Look for Emergent Behavior:**
   - ✓ Do new patterns appear from interactions?
5. **Evaluate System Goals:**
   - ✓ What is the purpose of the system?

5.Classroom Activity

Activity: Analyze a System Example

Steps:

1. Form small groups
2. Pick one example of a system (natural, human-made, organizational, or complex adaptive)
3. Identify **components, interconnections, feedback loops, and emergent behavior**
4. Draw a **system map or causal loop diagram**

5. Present findings to the class

Purpose:

- Helps students **connect theoretical concepts with practical systems**
- Reinforces understanding of **interconnections, feedback loops, and system behaviour**
- Prepares students for **analyzing complex systems in subsequent periods**

6.Best Practices

- Choose **representative examples relevant to students' experiences**
- Focus on **relationships and feedback, not just components**
- Identify **emergent behavior and system dynamics**
- Use **visual representation** to understand complexity

7.Common Mistakes

 Focusing only on components without interconnections
☐ Ignoring feedback loops and emergent behavior
☐ Oversimplifying complex systems
☐ Failing to link examples to real-world learning

8.Key Takeaways

- Real-world examples make **system thinking concrete and understandable**
- Systems can be **natural, human-made, organizational, or complex adaptive**
- Analyzing components, interconnections, feedback loops, and emergent behavior is crucial

- Visual tools like **system maps and causal diagrams** help analyze systems effectively

---

Period 4

Understandings of Systems

"Understandings of Systems = Knowledge of how systems behave, how components interact, and how to apply insights for effective decision-making."

1.What Does "Understandings of Systems" Mean?

Definition:

Understanding systems refers to the ability to **analyze components, relationships, feedback loops, and patterns** to gain insights that guide **effective problem-solving, design, and decision-making**.

Key Points:

- Goes beyond identifying components
- Focuses on **system behavior and dynamics**
- Helps **predict consequences of changes**
- Supports **holistic and sustainable solutions**

2.Key Insights in Understanding Systems

1. **Interconnectedness:**
    - ✓ Everything in a system is linked; a change in one part affects others

2. **Feedback Loops:**
    - ✓ Circular cause-and-effect mechanisms that **stabilize or amplify system behavior**
3. **Patterns and Trends:**
    - ✓ Systems often show recurring behaviors over time
4. **Emergence:**
    - ✓ Complex interactions can produce **unexpected behaviors** not seen in individual components
5. **Boundaries:**
    - ✓ Defining the system boundary helps **focus analysis and decision-making**
6. **Leverage Points:**
    - ✓ Small changes in key areas can **produce significant impact**

3.Why Understanding Systems is Important

- Enables **holistic problem-solving** instead of addressing only symptoms
- Helps **anticipate unintended consequences** of changes
- Supports **innovation by identifying opportunities within system dynamics**
- Facilitates **effective decision-making in complex environments**

4.Techniques to Understand Systems

Technique	Purpose	Example
Causal Loop Diagrams	Understand cause-effect and feedback	Map how classroom occupancy affects notifications and student movement
System Mapping	Visualize components and interconnections	Show students, classrooms, schedules, and app interactions

Technique	Purpose	Example
Stock and Flow Diagrams	Track accumulation and movement	Number of available classrooms (stock) changes as courses are scheduled (flow)
Scenario Analysis	Explore system behavior under different conditions	What happens if student enrollment increases by 20%?
Simulation Models	Test system dynamics virtually	Simulate peak classroom usage to predict congestion

5.Example – Engineering Scenario

System: Smart traffic light control system

Insights:

• **Insight 1:** Heavy traffic during peak hours causes long vehicle delays

• **Insight 2:** Real-time traffic sensors help adjust signal timings

• **Insight 3:** Combining sensor data with adaptive signal control improves traffic flow

Leverage Point:

• Adjusting traffic signal timing based on real-time data has a high impact on reducing congestion

Outcome:

Understanding the system helps design better traffic control strategies that reduce travel time and improve road efficiency.

6.Classroom Activity

Activity: Analyze System Understandings

Steps:

1. Form small groups
2. Choose a campus or daily life system

3. Identify **components, interconnections, feedback loops, and patterns**
4. Highlight **emergent behaviors and leverage points**
5. Present findings and discuss potential interventions

Purpose:

- Reinforces **insight-driven system analysis**
- Helps students **apply system thinking to real-world problems**
- Prepares students for **studying complex systems in subsequent periods**

7.Best Practices

- Focus on **interconnections, feedback loops, and patterns**
- Identify **emergent behaviors** for better understanding
- Look for **leverage points to implement effective changes**
- Use **visual tools** to clarify insights
- Think **long-term and system-wide**, not just short-term fixes

8.Common Mistakes

☐ Treating components in isolation
☐ Ignoring feedback loops and system dynamics
☐ Failing to identify leverage points
☐ Overlooking emergent behaviors and patterns

9.Key Takeaways

- Understanding systems involves **analyzing relationships, feedback, patterns, and behaviors**
- Insight into system dynamics supports **holistic problem-solving and effective decision-making**

- Techniques like **system maps, causal loops, stock & flow diagrams, and scenario analysis** are essential
- Identifying **leverage points** can maximize impact with minimal effort

---

Period 5

Complex Systems – Introduction

"Complex Systems = Systems with many interacting components, exhibiting emergent, adaptive, and often unpredictable behavior."

1. What are Complex Systems?

Definition:

A **complex system** is a system in which multiple components interact in ways that produce **emergent behaviors**, where the whole system is more than the sum of its parts.

Key Characteristics:

1. **Interconnected Components:** Many elements interacting dynamically
2. **Emergent Behavior:** Patterns arise that are **not obvious from individual components**
3. **Adaptation:** System components adjust based on interactions and environment
4. **Non-linear Relationships:** Small changes can lead to large effects
5. **Feedback Loops:** Reinforcing or balancing loops affect system behaviour

6. **Self-Organization:** Systems may spontaneously form order without central control

2.Examples of Complex Systems

Domain	Example	Components	Behavior Observed
Natural	Coral reef	Corals, fish, algae, water temperature	Dynamic balance, species interdependence, response to environmental changes
Social	Public transportation system	Buses, trains, passengers, routes, schedules	Peak-hour congestion, route adaptation, crowding patterns
Organizational	Hospital appointment system	Doctors, patients, rooms, scheduling software	Emergent patient flow patterns, adaptive rescheduling
Technological	Online shopping platform	Customers, products, recommendations, payment system	Buying trends, recommendation-driven behavior, rapid adaptation to promotions

3.Why Study Complex Systems?

- Helps understand **non-linear and adaptive behaviors**
- Allows for **better prediction and intervention** in real-world problems
- Supports **holistic and innovative problem-solving**
- Encourages **learning from patterns instead of only analyzing components**
- Useful in **engineering, management, environmental science, and social systems**

4.Key Concepts in Complex Systems

1. **Emergence:**
    - ✓ Behavior or patterns that cannot be predicted from individual components alone
2. **Non-linearity:**
    - ✓ A small change can produce a disproportionate effect
3. **Adaptation:**
    - ✓ Components evolve or adjust to changes in environment or other components
4. **Feedback Loops:**
    - ✓ Circular relationships that reinforce or stabilize system behavior
5. **Self-Organization:**
    - ✓ Components spontaneously form organized structures or behaviors
6. **Robustness and Fragility:**
    - ✓ Systems can be resilient to some disturbances but sensitive to others

5.Techniques to Study Complex Systems

1. **System Mapping:** Visualize interactions and relationships
2. **Causal Loop Diagrams:** Identify feedback loops and emergent patterns
3. **Simulation Models:** Test system behavior under different scenarios
4. **Agent-Based Modeling:** Model individual components and their interactions
5. **Scenario Analysis:** Explore consequences of changes or interventions

6.Classroom Activity

Activity: Identify a Complex System

Steps:

1. Form small groups
2. Choose a real-world system (e.g., campus, transportation, ecosystem, social media)
3. Identify **components, interactions, feedback loops, and emergent behaviors**
4. Discuss **how small changes can create large effects**
5. Present findings and patterns observed

Purpose:

- Helps students **recognize complexity in real-world systems**
- Encourages **holistic thinking and analysis of dynamic interactions**
- Prepares for **understanding system behaviors and interventions in subsequent periods**

7.Best Practices

- Focus on **relationships and interactions** rather than individual components
- Look for **emergent behaviors and adaptation patterns**
- Identify **feedback loops and leverage points**
- Use **visual and modeling tools** for better understanding
- Emphasize **long-term thinking and system-wide effects**

8.Common Mistakes

 Treating complex systems as simple linear systems

☐ Ignoring interactions and feedback loops

☐ Focusing solely on individual components

☐ Overlooking emergent or adaptive behaviors

9.Key Takeaways

- Complex systems consist of **many interacting components with adaptive, emergent, and non-linear behavior**
- Feedback loops, self-organization, and emergent patterns are crucial to understanding these systems
- Studying complex systems helps **anticipate consequences, design interventions, and solve real-world problems**
- Tools like **system maps, causal loops, simulations, and agent-based models** are essential for analysis

---

Period 6

Characteristics of Complex Systems

"Characteristics of Complex Systems = Features that distinguish complex systems from simple or linear systems, helping us understand their dynamic behavior."

1.Key Characteristics of Complex Systems

1. Interconnected Components

- A complex system is made up of **multiple components that interact with each other**.
- Example: In a university system, students, faculty, classrooms, schedules, and notifications interact.

2. Emergence

- **Emergent behavior** arises from interactions between components and cannot be predicted by looking at components individually.
- Example: Traffic jams emerge from the interactions of vehicles, drivers, and signals, not from any single vehicle.

3. Adaptation

- Components of the system **adjust their behavior based on changes in the system or environment**.
- Example: Students adapt their schedules when classroom availability changes.

4. Non-linearity

- Small changes in one part of the system can produce **large and sometimes unexpected effects**.
- Example: Adding one extra class in a busy classroom may cause congestion across multiple floors.

5. Feedback Loops

- Circular cause-and-effect relationships that can **reinforce or stabilize system behavior**.
- **Positive feedback:** Amplifies change (e.g., more students using an app → faster adoption)
- **Negative feedback:** Balances or stabilizes the system (e.g., scheduling limits prevent overcrowding)

6. Self-Organization

- Complex systems can **form organized structures or behaviors without central control**.
- Example: Birds flocking or students naturally forming study groups without a formal plan.

7. Dynamic and Time-Dependent Behavior

- System behavior **changes over time**, often in unpredictable ways.
- Example: Classroom occupancy patterns fluctuate throughout the day based on courses and student movement.

8. Sensitivity to Initial Conditions

- Small differences in starting conditions can lead to **very different outcomes**.
- Example: Early enrollment patterns affect classroom congestion throughout the semester.

9. Emergent Patterns

- Repeating patterns or trends emerge from complex interactions.
- Example: Peak-hour congestion patterns in campus hallways or online platform usage spikes.

2. Example: Supermarket Checkout System

A supermarket checkout system is a **complex system** because it is made up of many parts that interact with each other in dynamic ways.

- **Interconnected Components:**
  Customers, cashiers, and checkout counters are connected. When

more customers arrive, queues become longer, which affects waiting time for everyone.

- **Emergence:**
  No one plans how queues will look, but orderly lines and congestion naturally form as customers choose counters. These patterns emerge from individual decisions.
- **Adaptation:**
  Customers adapt by switching to shorter queues or choosing self-checkout when lines are long.
- **Non-linearity:**
  Adding just one customer to a busy counter can suddenly increase waiting time a lot, showing that small changes can have big effects.
- **Feedback Loops:**
  Long queues signal the store to open more counters (negative feedback to reduce waiting). If queues are short, fewer counters are opened.
- **Self-Organization:**
  Customers automatically form lines without central control or instructions.
- **Dynamic Behavior:**
  Queue lengths constantly change depending on time of day, promotions, or holidays.
- **Sensitivity:**
  A small delay, such as a price check for one item, can affect many customers waiting behind.
- **Emergent Patterns:**
  Predictable peak congestion appears during evenings or weekends when more people shop.

3.Classroom Activity

Activity: Identify Characteristics in a System

Steps:

1. Form small groups
2. Choose a real-world system (campus, traffic, ecosystem, online platform)
3. Identify **components and interactions**
4. List **observed characteristics of a complex system** (emergence, feedback, adaptation, etc.)
5. Present findings and discuss implications for managing or improving the system

Purpose:

- Reinforces **understanding of complex system characteristics**
- Helps students **apply concepts to real-world scenarios**
- Prepares students for **analyzing system interventions and design strategies**

4.Best Practices

- Focus on **interactions and emergent behaviors**
- Look for **feedback loops and adaptation mechanisms**
- Recognize **non-linear and dynamic behavior**
- Use **visual tools** like causal loops and system maps to understand complexity
- Consider **sensitivity to initial conditions** when planning interventions

5.Common Mistakes

 Treating complex systems as simple linear systems
☐ Ignoring feedback loops or emergent behavior
☐ Focusing only on individual components
☐ Overlooking dynamic behavior over time

6.Key Takeaways

- Complex systems are **dynamic, adaptive, and interconnected**
- Key characteristics include **emergence, feedback loops, non-linearity, self-organization, and sensitivity to initial conditions**
- Understanding these characteristics is essential for **system analysis, problem-solving, and innovative solution design**
- Classroom exercises help students **recognize and apply these concepts in real-world systems**

Period 7

Examples of Complex Systems

"Examples of Complex Systems = Real-world systems with multiple interacting components, emergent behaviors, and adaptive dynamics."

1.Importance of Studying Examples

- Makes **abstract concepts tangible**
- Helps **recognize patterns and feedback loops**
- Demonstrates **non-linear and adaptive behaviors**
- Encourages **holistic thinking for problem-solving and innovation**

2.Examples of Complex Systems Across Domains

Domain	Example	Components	Emergent Behavior / Observations
Natural	Forest ecosystem	Trees, animals, insects, soil, rainfall	Natural regeneration, species balance, food-chain dynamics
Social	Urban public transport	Buses, trains, passengers, routes, schedules	Crowding patterns, peak-hour congestion, route shifting
Organizational	Manufacturing plant	Machines, workers, production schedules, inventory	Bottlenecks, workflow optimization, adaptive production rates
Technological	Internet network	Servers, routers, users, data packets	Network congestion, load balancing, self-healing routes
Economic	Supply chain system	Suppliers, manufacturers, distributors, retailers	Demand–supply fluctuations, bullwhip effect
Healthcare	Emergency response system	Ambulances, hospitals, staff, communication systems	Response-time patterns, resource strain during emergencies
Environmental	River system	Rainfall, rivers, dams, ecosystems	Flood cycles, sediment movement, seasonal flow changes

3. Example

System: Water distribution system in a residential area

Components: Water source, pumps, pipes, storage tanks, households

Interactions:
Household water usage affects pressure; pump operation responds to demand changes

Emergent Patterns:
Low pressure during peak usage times, balanced flow during off-peak hours

Observation:
Understanding the system helps design better pump scheduling and storage capacity to ensure reliable water supply.

4.How to Analyze Examples

1. **Identify Components:** Determine key elements of the system
2. **Observe Interactions:** How components influence each other
3. **Look for Emergent Behavior:** Identify patterns that arise from interactions
4. **Identify Feedback Loops:** Positive or negative cycles affecting system behavior
5. **Recognize Adaptation:** How the system adjusts to changes or disturbances

5. Classroom Activity

Activity: Identify Complex Systems Around You

Steps:

1. Form small groups

2. Pick a real-world system (campus, ecosystem, social media, traffic, hospital)
3. Identify **components, interactions, feedback loops, and emergent behaviors**
4. Discuss how small changes can **affect the system as a whole**
5. Present findings to the class

Purpose:

- Reinforces **application of complex system concepts**
- Helps students **recognize adaptive and emergent behaviors**
- Prepares students for **analyzing system interventions in subsequent periods**

6.Best Practices

- Choose **systems students are familiar with** for easier understanding
- Focus on **interactions and feedback rather than isolated components**
- Highlight **emergent behaviors and adaptive patterns**
- Use **visual mapping or diagrams** to represent the system
- Discuss **cause-and-effect relationships** and potential interventions

7.Common Mistakes

 Focusing only on components without interactions
☐ Ignoring feedback loops or emergent patterns
☐ Oversimplifying complex systems
☐ Not recognizing adaptive behaviors

8.Key Takeaways

- Complex systems are **everywhere – natural, social, organizational, technological, economic, and environmental**
- Recognizing components, interactions, feedback loops, and emergent behaviors is essential
- Examples help students **connect theory to practical applications**
- Analyzing real-world systems prepares students for **problem-solving, innovation, and strategic decision-making**

Period 8

Applications of System Thinking

"Applications of System Thinking = Using a holistic approach to analyze and improve systems in practical scenarios."

1.Why System Thinking is Applied

- Provides a **holistic understanding** of complex problems
- Helps **identify root causes** instead of just addressing symptoms
- Predicts **long-term consequences** of interventions
- Improves **decision-making and problem-solving**
- Supports **innovation, efficiency, and sustainability**

2.Domains of System Thinking Applications

Domain	Application	Example / Explanation
Business & Management	Process optimization	Analyze workflows in a company to reduce delays and increase productivity
Healthcare	Patient care management	Optimize hospital operations, patient flow, and resource

Domain	Application	Example / Explanation
		allocation
Education	Campus management	Improve classroom scheduling, resource utilization, and student engagement
Environmental & Ecology	Sustainable resource management	Understand ecosystems to balance conservation, resource use, and human activity
Technology & IT	Software development & system integration	Analyze dependencies and interactions between different software modules or platforms
Government & Policy	Urban planning & policy-making	Evaluate city infrastructure, traffic systems, and public services holistically
Social Systems	Community development	Plan social programs considering interactions between stakeholders, resources, and outcomes

3.Examples

1. **Campus Classroom Allocation System**
   - ✓ Components: Classrooms, students, faculty, notifications, schedules
   - ✓ System Thinking Application: Analyze interconnections and feedback loops to **optimize classroom usage and reduce congestion**
2. **Traffic Management System**
   - ✓ Components: Roads, vehicles, traffic lights, pedestrians
   - ✓ System Thinking Application: Identify bottlenecks, predict traffic flow patterns, and plan interventions for **smoother movement**
3. **Hospital Resource Allocation**
   - ✓ Components: Patients, doctors, equipment, wards
   - ✓ System Thinking Application: Optimize patient flow, staff allocation, and equipment use for **better healthcare outcomes**

4. **Software Development Project**
    - ✓ Components: Developers, modules, dependencies, users
    - ✓ System Thinking Application: Analyze interactions between modules to **reduce bugs, integration issues, and delays**

4.Techniques for Applying System Thinking

1. **System Mapping** – Visualize components and relationships
2. **Causal Loop Diagrams** – Understand feedback loops
3. **Stock & Flow Analysis** – Track resource accumulation and movement
4. **Scenario Analysis** – Predict outcomes under different conditions
5. **Simulation Models** – Test interventions virtually before implementation

5.Classroom Activity

Activity: Applying System Thinking to a Real Problem

Steps:

1. Form small groups
2. Select a real-world problem (campus, hospital, business, traffic)
3. Identify **components, interactions, feedback loops, and system boundaries**
4. Suggest **possible interventions using system thinking principles**
5. Present insights to the class

Purpose:

- Reinforces **practical application of system thinking**

- Encourages students to **think holistically and propose sustainable solutions**
- Prepares students for **analyzing complex systems in real-world contexts**

6.Best Practices

- Focus on **relationships and interdependencies**
- Analyze **root causes, not just symptoms**
- Consider **short-term and long-term impacts**
- Use **visual tools** for clarity
- Encourage **collaborative problem-solving**

7.Common Mistakes

☐ Treating problems in isolation
☐ Ignoring feedback loops and emergent behaviors
☐ Failing to define system boundaries
☐ Overlooking long-term consequences

8.Key Takeaways

- System thinking **can be applied in business, healthcare, education, technology, social systems, and government**
- Helps **analyze problems holistically, identify root causes, and design effective interventions**
- Visual tools like **system maps, causal loops, and simulations** aid practical application
- Classroom exercises help students **translate theory into actionable solutions**

Period 9

Case-Based Understanding of Complex Systems

"Case-Based Understanding = Applying system thinking to real-world examples to analyze complexity, identify interactions, and propose solutions."

1.Purpose of Case-Based Learning

- Connects **theory to real-life applications**
- Demonstrates **interconnections, feedback loops, and emergent behaviors** in real systems
- Develops **analytical, problem-solving, and decision-making skills**
- Encourages **collaborative learning and discussion**

2.Steps for Case-Based Analysis

1. **Identify the System** – Define components, boundaries, and purpose
2. **Map Interconnections** – Understand relationships between components
3. **Analyze Feedback Loops** – Identify reinforcing and balancing loops
4. **Observe Emergent Patterns** – Look for behaviors not visible in individual components
5. **Identify Leverage Points** – Determine where small interventions can produce large effects
6. **Propose Interventions** – Suggest improvements or solutions based on system understanding

3.Case Study Examples

Case 1: University Campus Classroom Management

- **System Components:** Classrooms, students, faculty, schedules, notifications
- **Observed Complexity:** Peak-hour congestion, adaptive student scheduling, informal study groups
- **Feedback Loops:** Notifications reduce congestion → student behavior adapts → balanced classroom usage
- **Emergent Patterns:** Classroom occupancy peaks emerge despite individual scheduling plans
- **Interventions:** Adjust notifications, stagger class schedules, optimize room allocation

Case 2: Hospital Patient Flow System

- **System Components:** Patients, doctors, nurses, equipment, wards
- **Observed Complexity:** Bottlenecks during peak patient intake, delayed treatments
- **Feedback Loops:** Staff workload affects patient processing → delays increase → patient outcomes affected
- **Emergent Patterns:** Queue buildup and resource scarcity
- **Interventions:** Allocate resources dynamically, prioritize critical patients, streamline workflow

Case 3: Traffic Management System

- **System Components:** Roads, vehicles, traffic signals, pedestrians, public transport
- **Observed Complexity:** Non-linear congestion patterns, unpredictable traffic jams

- **Feedback Loops:** More vehicles → slower movement → traffic signal adjustments → route changes
- **Emergent Patterns:** Peak-hour congestion, adaptive driver behavior
- **Interventions:** Intelligent traffic lights, alternate routes, public transport incentives

4. Classroom Activity

Activity: Case-Based Group Analysis

Steps:

1. Divide students into groups
2. Assign one real-world system per group (campus, hospital, traffic, social media, ecosystem)
3. Ask each group to:
    - ✓ Identify system components, boundaries, and purpose
    - ✓ Map interconnections and feedback loops
    - ✓ Identify emergent behaviors and patterns
    - ✓ Suggest leverage points and possible interventions
4. Present analysis to the class

Purpose:

- Reinforces **system thinking concepts in real-world contexts**
- Encourages **critical thinking, collaboration, and solution-oriented discussion**
- Prepares students for **designing interventions in applied innovation and design projects**

5. Best Practices

- Choose systems relevant to students' daily experiences
- Encourage visual mapping of components and interactions
- Focus on interconnections, feedback loops, and emergent behaviour
- Discuss both short-term and long-term impacts of interventions

6.Common Mistakes

☐ Focusing only on individual components
☐ Ignoring emergent behaviors or feedback loops
☐ Proposing interventions without analyzing system dynamics
☐ Overlooking system boundaries and scope

7.Key Takeaways

- Case-based learning helps **connect theory to practice**
- Real-world systems demonstrate **complexity, interconnections, feedback loops, and emergent patterns**
- Identifying **leverage points** enables effective interventions
- Collaborative analysis develops **critical thinking, problem-solving, and innovation skills**
