Grade Four Computer Science

Full Unit

The Consortium
Alberta Professional Learning Consortium

Grade 4 CS Teacher Notes

This unit provides a comprehensive, design-focused curriculum for Grade 4 students. It is built on the core idea that **Design** involves processes that transform ideas into artifacts—both physical and digital—to meet specific needs. Each chapter includes force copy links to student notes, worksheets and full lesson plans.

Unit Overview

The curriculum is divided into an introduction, six core chapters, and a concluding final project. Each section focuses on a different aspect of computer science and design thinking:

Introduction: Defines design, needs, and the basic four-step design process.
Chapter 1: Deep dive into the seven iterative steps of the **Design Thinking Process**.
Chapter 2: Focuses on the power of **Feedback** and the iterative nature of design.
Chapter 3: Explores different types of **Artifacts**, such as models, blueprints, and digital programs.
Chapter 4: Applies design thinking to **Complex World Problems**, featuring case studies on mobile medical clinics and self-driving cars.
Chapter 5: Covers practical **Design Considerations**, specifically the availability and cost of materials.
Chapter 6: Introduces **Algorithms** as precise, step-by-step instructions (the "recipes" for computers).

How to Use This Unit

Each chapter follows a flexible, three-step lesson structure designed to support differentiated instruction and varied classroom technology access.

Step 1: Guided or Self-Guided Learning

Begin each chapter by introducing the core concepts. You can guide the class together or allow students to choose how they consume the information through the following provided resources:

Primary Text: Student notes and chapter content.
Multimedia: Links to videos, podcasts, and infographics.
Check for Understanding: Use the included **Student Worksheets** (available for print or digital use) to assess initial comprehension.

Step 2: Activity Selection (Differentiated Application)

Choose one (or more) of the following activity paths based on your classroom resources and student interests:

Unplugged: Collaborative, low-tech activities focusing on critical thinking.

Micro:bit: Introduction to hardware and physical computing using block coding.
Scratch: Software development and digital artifact creation through visual coding.

Step 3: Assessment

Conclude each chapter with a digital quiz. The unit culminates in a **Final Project** and a comprehensive test, allowing students to demonstrate their mastery of the design process.
**Note you will need to use the folder copy link provided above to access quizzes/tests.*

Key Resource Repository

Each chapter in the Unit Plan contains direct Google Drive template links to:
Student Notes: Detailed content summaries for student reference.
Lesson Plans: Step-by-step guides for teachers.
Worksheets: Differentiated tasks for self-guided or unplugged learning.

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 Computer Science Unit Plan
Copy Grade 4 Computer Science Unit Overview
Copy Grade 4 Computer Science Activity Lesson Plans
Copy Grade 4 Computer Science Materials List
Copy Grade 4 Computer Science Curriculum
Open Form: Grade 4 Computer Science Final Unit Exam: Design & Algorithms

Grade 4 CS Materials List

Technology & Hardware

These items are necessary for the core digital and physical computing activities within the unit, including programming and testing.

Item	Description/Notes
Chromebooks	Must have reliable internet access for online tools.
micro:bit V2	**Preferred** for built-in sound and vibration features crucial for accessibility testing.
MakeCode Editor	Online web version or desktop application is acceptable for programming micro:bit devices.
Scratch	Online web version or desktop version for introductory programming and digital prototyping.

"Maker Bin" & Craft Supplies

Building Materials

Material	Use
Cardboard	General construction and structural integrity.
Cardboard Tubes	Creating rollers, joints, or rounded structures.
Index Cards	Small structural elements, signs, and prototyping smaller features.
Straws	Lightweight supports and movement mechanisms.
Pipe Cleaners	Flexible connectors and detail work.

Fasteners & Tools

Item	Use
Masking Tape	General fastening; easily removable for iteration.
Rubber Bands	Creating tension, simple mechanisms, and temporary connections.
String	Pull mechanisms, measuring, and hanging/securing.
Clothespins	Temporary clamps and simple grips.
Scissors	Cutting and shaping all paper and cardboard materials.
Markers	Labeling, decorating, and adding visual detail to prototypes.

Classroom Props & Testing Gear

These items are used in hands-on activities to build empathy and test prototypes under simulated accessibility constraints.

Item	Purpose
Physical Props	Testing how prototypes interact with daily items: a jacket, hat, backpack, ball or block, and a sweatshirt.
Blindfolds	Used to simulate low or no vision for accessibility testing (clean sleep masks can be used as an alternative).
Scarves	Used to tie one arm back to simulate limited mobility/use of one arm.
Mystery Items	The "Mystery Box" activity for sensing and descriptive language.
Cardboard Box with a Hole	Used as a "feely box."
Common Objects	Stapler, spoon, toy, and photo for identification by touch.

Stationery & Documentation

Materials essential for planning, recording ideas, and organizing the unit's final deliverables.

Item	Description
Large Paper	For collaborative design mapping, brainstorming, and mind-mapping.
Plain Paper	Used to create individual "Designer Notebooks" for sketching and ideation.
Worksheets	Scavenger Hunt sheets, Blueprint worksheets, and "Secret Code Key" cards
Organization	A folder for each student's "Final Project Portfolio" to contain all documentation and deliverables.

Welcome to the World of Design! 💡

Have you ever looked at something and thought, "Someone had to make that"? 🤔 From your phone to your chair to the clothes you are wearing, everything around you was once just an idea! Design is the amazing process that takes those ideas and turns them into real things that help us.

In this book, we're going to explore how we move from a simple thought to a finished *artifact*—a word we use for something that has been made or designed.

What is Design?

Design is a focused process of planning and making something. It's not just about drawing a picture; it's about solving a problem or meeting a need.

The Core Idea: Needs

Every great design starts with a **need**. A need is a problem that requires a solution.

Need Example	Problem to Solve
Staying dry in the rain	Water falls on you
Carrying heavy books	Your arms get tired
Talking to a friend far away	You can't hear them or see them

When you design something, you are creating an artifact specifically to satisfy one of these needs.

From Idea to Artifact: The Design Process

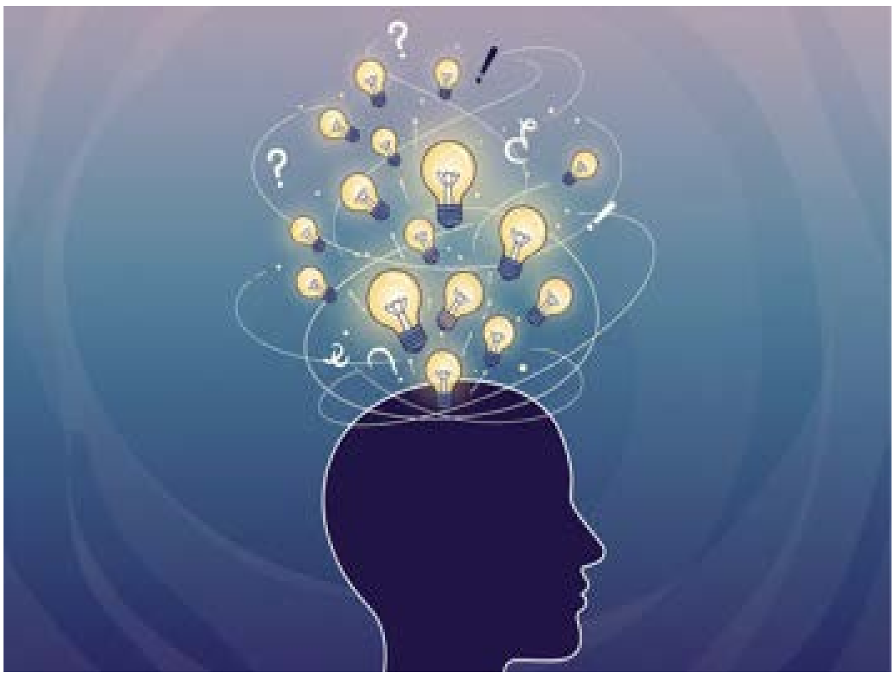
The journey from an idea to a finished artifact involves several steps. Think of it like a recipe 🍳 for making something new! This journey is what we call the **Design Process**.

Step 1: Idea Generation (The Spark) ✨

Every artifact begins as an idea. This is the moment you figure out *what* you are going to make to solve the need.

- **Brainstorming:** Thinking of many possible solutions.
- **Sketching:** Drawing simple pictures to quickly show your ideas.

It's okay if your first idea isn't perfect! Designers try out lots of different ideas before picking the best one.



Step 2: Planning and Prototyping (The Blueprint) 📝

Once you have a strong idea, you need a plan. This is where you figure out the details:

- **Materials:** What will it be made of (plastic, wood, code)?
- **Size and Shape:** How big will it be?
- **Steps:** What is the order of steps needed to build it?

A **prototype** is an early, simple model of your artifact. It's a test version! You might build a small model out of cardboard or program a simple version on a computer 💻.

Step 3: Creation (The Build) 🛠️

This is the step where you transform your plan into the actual artifact. Using your chosen materials and following your steps, you build the real thing. This step requires focus and attention to detail.

Step 4: Testing and Evaluating (The Check) ✅

Does your artifact actually meet the need you started with? This is the time to find out.

You test the artifact to see if it works as planned.
If you designed a new chair, you would sit in it to test its strength and comfort.
If you programmed an app, you would use it to find bugs (errors in the code).

Often, testing shows that the artifact needs changes. This is normal! Good design is an ongoing cycle.

Artifacts: The Final Outcome

An **artifact** is the final product of the design process. It is the physical (like a spoon 🥄) or digital (like a video game 🎮) creation that transforms your initial idea into something real and useful.

Design ensures that these artifacts are:

Functional: They work and do what they are supposed to do.
Useful: They actually meet the identified need.
Usable: They are easy and simple for people to use.

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Introduction: Welcome to the World of Design

Copy Grade 4 CS Introduction Lesson Plan

View Grade 4 CS Introduction_Video

View Grade 4 CS Introduction Podcast

Copy Grade 4 CS Introduction Slides

View Grade 4 CS Introduction Infographic

Copy CS Scavenger Hunt & Robot Coding

Copy Design Detective Worksheet: From Need to Artifact! 💡

Introduction to the Design Thinking Process

Design thinking is a powerful, non-linear process that teams use to understand users, challenge assumptions, redefine problems, and create innovative solutions. It is not a fixed, step-by-step methodology but rather a framework that encourages continuous learning and iteration.

The core of the design thinking process can be broken down into the following key steps:

1. Understanding the Problem (Empathize & Define)

The first and most critical step is gaining a deep, empathetic understanding of the people you are designing for and the specific problem you are trying to solve. This involves more than just gathering data; it requires seeing the world through the users' eyes.

Component	Description	Example
Empathize	Conduct research (interviews, observations) to gain an empathetic understanding of the users' needs and motivations.	A design team observes commuters using public transit to understand their pain points during the morning rush.
Define	Synthesize the research findings to clearly articulate the core problem. This is often framed as a "Point-of-View" statement.	*Problem Statement:* "Bus commuters need a way to easily check real-time bus locations because the current schedule displays are often inaccurate, causing anxiety about being late."

2. Forming Ideas (Ideating)

With a clear problem definition, the next phase is to brainstorm a wide range of creative solutions. The goal here is quantity over quality—to explore all possibilities, no matter how unconventional they may seem.

Techniques for Ideation

- **Brainstorming:** A group method where all participants contribute ideas without judgment.
- **Mind Mapping:** Visually organizing information to show relationships between different ideas.
- **Sketching/Worst Possible Idea:** Generating initial low-fidelity concepts or, conversely, brainstorming the *worst* possible solutions to break creative blocks.

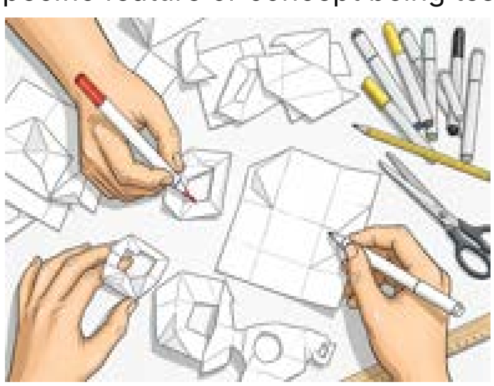
3. Planning

Once a set of promising ideas has been generated, the team moves into the planning phase. This involves selecting the most viable concept(s) and outlining the steps needed to bring them to life.

Task	Outcome	Tool/Resource
Select the Best Idea	Decide on the solution(s) to move forward with, based on user needs, feasibility, and business goals.	Decision Matrix/Scoring Grid
Develop a Strategy	Define the scope, resources, and timeline for creating the prototype.	Project Roadmap
Identify Key Features	Determine the essential elements that must be included in the first version.	Feature List

4. Creating (Prototyping)

The creation phase is about turning ideas into tangible forms—called prototypes—that can be tested. A prototype can be anything from a paper model to a basic digital mockup. The purpose is to allow the team to fail fast and learn quickly.**Key Principle:** The prototype should be low-cost and quick to produce, focusing only on the specific feature or concept being tested.



5. Analyzing (Evaluating Outcomes)

Before testing, or during an initial review of the prototype, the team analyzes the design against the original problem statement and identified needs. This is an internal check to ensure the solution aligns with the goals.

Questions for Internal Analysis:

- Does this prototype directly address the user's defined need?
- Is it technically feasible with the available resources?
- Does it offer a clear, intuitive user experience?
- What assumptions have we made that need validation?

6. Testing

Testing involves putting the prototype in front of real users to gather feedback. This step is crucial for uncovering flaws and learning how the solution performs in a realistic context.

The testing event, often called a usability test, is an opportunity to validate or disprove the assumptions made in earlier stages. Make sure to schedule the event here: Calendar event

7. Troubleshooting (Iterate)

Based on the feedback from testing, the team enters the final, but continuous, phase: troubleshooting and iteration. Design thinking is non-linear, meaning the results of the testing phase often send the team back to an earlier step—be it ideating new solutions, refining the problem definition, or simply creating a new, improved prototype.

The process repeats until a satisfactory solution is found and implemented at the target location, such as a major transit hub in Place.

Test Result	Action Required (Troubleshooting)
Users found a key feature confusing.	Go back to **Forming Ideas** and brainstorm clearer ways to present the feature.
The prototype solved the stated problem but revealed a new, deeper user need.	Go back to **Understanding the Problem** (Empathize & Define) to redefine the scope.
The technical implementation was too costly.	Go back to **Planning** to adjust the scope or find more cost-effective technologies.

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Chapter 1: Introduction to the Design Thinking Process

Copy Grade 4 CS Chapter 1 Lesson Plans

Copy Grade 4 CS Chapter 1 Self Guided Learning Worksheet

View Grade 4 CS Chapter 1 Podcast

View Grade 4 CS Chapter 1 Infographic

View Grade 4 CS Chapter 1 Video

Copy Grade 4 CS Chapter 1 Slides

Copy Grade 4 CS Chapter 1 micro:bit Worksheet

Copy Grade 4 CS Chapter 1 Scratch Worksheet

Copy Grade 4 CS Chapter 1 Unplugged Worksheet

Open Form: Grade 4 CS Chapter 1 Quiz

The Power of Feedback 💡

Welcome, Grade 6 learners! Today we are going to explore one of the most important parts of designing and creating anything—**feedback**! Whether you are building a website, designing a new game, or creating an app, getting feedback is the secret sauce that makes your project better.

What is Feedback?

Feedback is simply information about how well something is working. When you show your design to another person, their reaction and comments are the feedback. It tells you what is great, what is confusing, and what might need to change so that your design meets everyone's needs.

Action	Example of Feedback
Showing a friend your game idea	"I love the characters, but the controls are hard to use."
Asking a teacher about your poster design	"The colors are bright, but the text is too small to read from far away."
Testing a new piece of software	"When I click this button, the program crashes." 💥

Why Do We Need Feedback in Design?

Imagine you are designing a new type of school desk. You might think, "I need a flat top and a place to put books." That is a great start! But what if you forget about left-handed students? What if you forget that some students use wheelchairs and need a desk at a specific height?

This is where feedback comes in. When you ask other people about your desk design, they might say:

- "I'm left-handed, and the pencil holder is on the wrong side for me."
- "This desk is too low for me to roll my wheelchair under it."
- "There isn't a hook to hang my backpack."

Feedback helps you see your design through different eyes. It makes sure you consider **all needs**—not just your own—to create the best possible product.

Considering All Needs 🤔

In the world of computer science and design, we often talk about **usability** (is it easy to use?) and **accessibility** (can *everyone* use it?).

Usability

Usability is about making sure a product is efficient and easy for its intended users. If a website takes five minutes to load, or if a video game has confusing instructions, the usability is poor.

- **Example:** A designer creates an app button that is so small a user with thick fingers cannot tap it accurately.
  - **Feedback:** "The button is too small, I keep missing it."
  - **Fix:** The designer makes the button bigger, improving usability.

Accessibility

Accessibility means designing something so that people with different abilities can still use it. This is a crucial area where feedback ensures all needs are considered.

- **Example:** A website uses only color to show if a piece of information is important (red means "stop," green means "go"). A person who is colorblind might not be able to understand this.
  - **Feedback:** A user who is colorblind points out they can't tell the difference between the colors.
  - **Fix:** The designer changes the design to use both color and symbols (like a stop sign shape 🛑 and a checkmark shape ✅) so everyone can understand the message. This improves accessibility.

The Feedback Loop 🔁

The design process is not just a straight line from "Idea" to "Finished Product." It's a loop! You use feedback to keep improving.

1. **Design:** You create your first version (a prototype or sketch).
2. **Test:** You share your design with others.
3. **Get Feedback:** You listen to what they say.
4. **Revise:** You use the feedback to change and improve your design.
5. **Test Again:** You go back to step 2 with your improved version.

You repeat this loop until the feedback tells you that the product is great and meets everyone's needs!

Remember, getting feedback is not about being wrong; it's about making your design stronger and more inclusive! Happy designing! 💻

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

   Copy Grade 4 CS Chapter 2: The Design Process: The Power of Feedback

   Copy Grade 4 CS Chapter 2 Lesson Plans

   Copy Grade 4 CS Chapter 2 Slides

   View Grade 4 CS Chapter 2 Podcast

   View Grade 4 CS Chapter 2 Infographic

   View Grade 4 CS Chapter 2 Video

   Copy Grade 4 CS Chapter 2 micro:bit Worksheet

   Copy Grade 4 CS Chapter 2 Unplugged Worksheet

   Copy Grade 4 CS Chapter 2 Self Guided Learning Worksheet

   Copy Grade 4 CS Chapter 2 Scratch Worksheet

   Open Form: Grade 4 CS Chapter 2 Quiz

What is an Artifact? 🤔

An **artifact** is simply anything that is made or created. It is an object or product that comes into existence through a process called **design**.

Who creates these artifacts?

- **Humans:** When you build a Lego model, write a story, or draw a picture, you are creating an artifact!
- **Machines:** Factories use machines to create thousands of artifacts every day, like cars 🚗, cell phones 📱, and toys.
- **Computers:** When you write code for a video game or a website, the code itself is an artifact, and so is the finished program! 💻

The Design Process

The process of design is a creative way to solve problems and create something new. Everything you see around you—from your chair to your school building—was created using some kind of design process.

When someone designs something, they are thinking about what is needed and how to make the best possible solution. The final result of this thinking and planning is the **artifact**.

Examples of Artifacts from Design 💡

Design is a huge field! It doesn't just produce physical objects you can touch. It also produces digital and conceptual artifacts.

Here are some different types of artifacts that come from the design process:

1. Objects

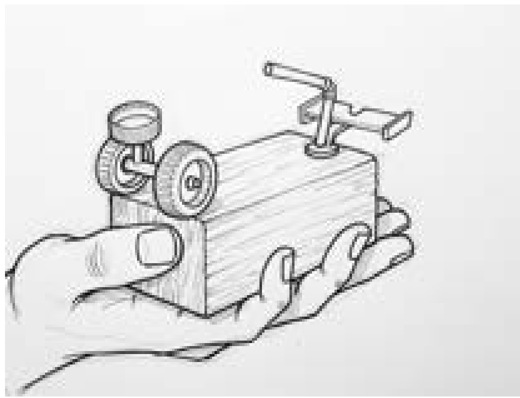
These are the most common artifacts. They are physical things you can hold and use.

Artifact Type	Description	Example
Objects	Physical products made for a specific purpose.	A specialized water bottle 💧 designed to keep drinks cold for 24 hours.

2. Prototypes

A **prototype** is a first or early model of a product. It is often created to test an idea and see if it works before spending a lot of time and money on the final product.

Artifact Type	Description	Example
Prototypes	A sample created to test a design or concept.	A small cardboard model of a new type of chair before the company builds it out of metal.



3. Models

A model is a representation of a real-world system or object. Models help people understand things that are too big, too small, or too complex to see easily.

Artifact Type	Description	Example
Models	A representation used to understand a concept or object.	A scale model of the solar system showing the planets 🪐 and the sun.

4. Blueprints

A **blueprint** is a detailed plan or design drawing used to make a building, machine, or other object. Think of it as a map for construction!

Artifact Type	Description	Example
Blueprints	Detailed technical drawings that serve as a plan.	Architectural plans showing all the measurements and materials needed to build a house 🏡.

Artifacts in Computer Science 🧑‍💻

Computer science relies heavily on design to create digital artifacts.

5. Algorithms

An **algorithm** is a set of step-by-step instructions designed to perform a specific task. Algorithms are the "recipes" that tell a computer what to do. They are conceptual artifacts, meaning you can't touch them, but they are designed and created.

Artifact Type	Description	Example
Algorithms	A step-by-step procedure for solving a problem or completing a task.	The steps a computer follows to sort a list of names alphabetically 🔠.

6. Programs

A **program** is a collection of algorithms written in a computer language (like Python or JavaScript). The final software or app you use is a program. The lines of code that make up the program are also artifacts.

Artifact Type	Description	Example
Programs	A set of coded instructions that tell a computer how to run.	A video game 🎮 or the app you use to check the weather ☁️.

7. Experiments

In design, especially when testing new ideas, **experiments** are designed to see how something behaves or if an idea works. The plan for the experiment and the results gathered are also artifacts of the design process.

Artifact Type	Description	Example
Experiments	A procedure carried out to test a hypothesis or discover a principle.	A structured test to see which color button 🟥 or 🟦 users click on most often on a website.

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Chapter 3: Artifacts & Design

Copy Grade 4 CS Chapter 3 Lesson Plans

View Grade 4 CS Chapter 3 Podcast

View Grade 4 CS Chapter 3 Video

Copy Grade 4 CS Chapter 3 Slides

View Grade 4 CS Chapter 3 Infographic

Copy Grade 4 CS Chapter 3 Self Guided Learning Worksheet

Copy Grade 4 CS Chapter 3 Scratch Worksheet

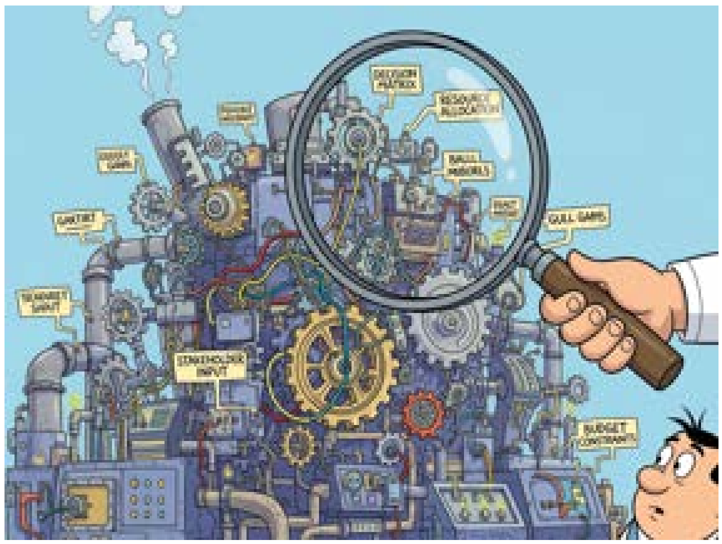
Copy Grade 4 CS Chapter 3 micro:bit Worksheet

Copy Grade 4 CS Chapter 3 Unplugged Worksheet

Open Form: Grade 4 CS Chapter 3 Quiz

Design Thinking: Solving Big World Problems 🌎

You already know that design helps us solve problems, like designing a better backpack or the best layout for a website. But the design process is so powerful that it's used to solve some of the biggest and toughest challenges in the world! This is called using **Design Thinking** to deal with **complex problems**.

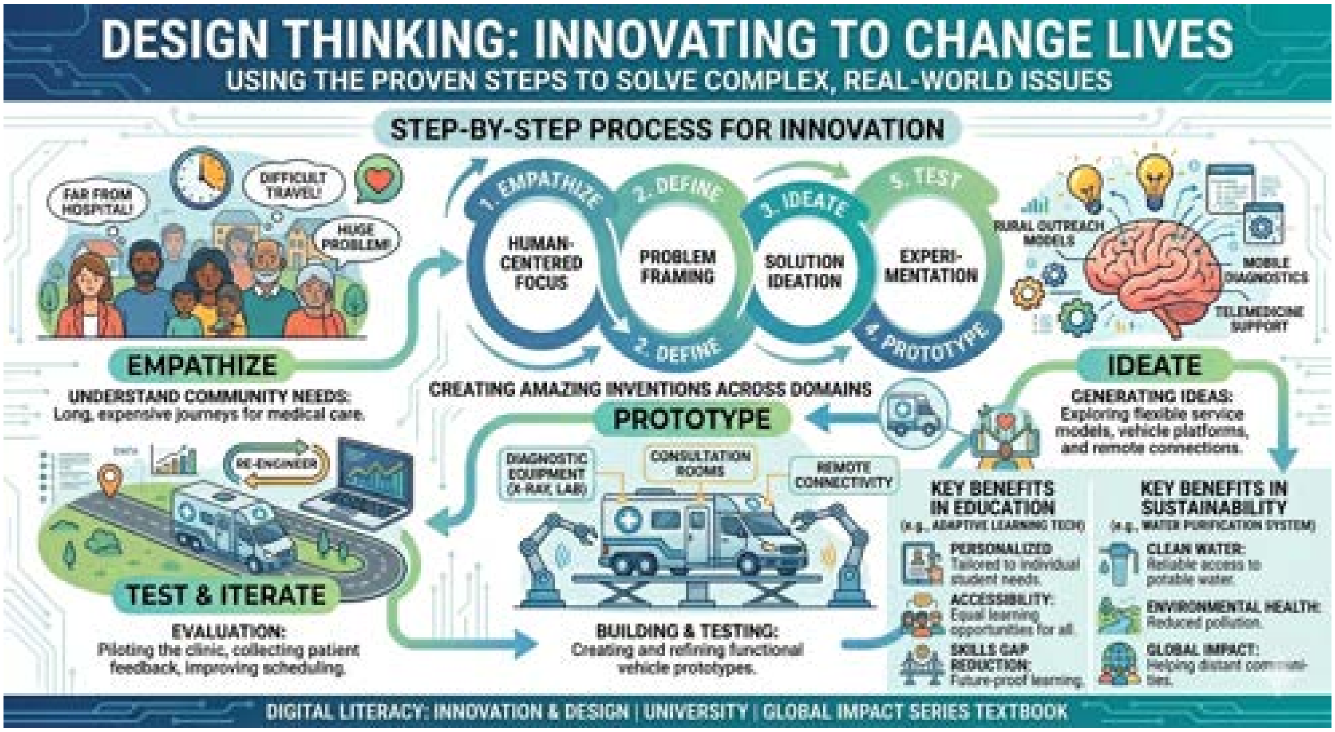


What is a Complex Problem? 🤔

Remember, a **complex problem** is a challenge that is difficult to solve because it has many connected parts, no single clear answer, and affects many different people.

Simple Problem	Complex Problem
Fixing a broken toy	Designing a robot to help the elderly
Writing a short thank-you note	Creating a social media platform that is safe for everyone
Choosing your favorite color	Designing a power source that doesn't harm the planet

When we use Design Thinking for these complex, real-world issues, we can create amazing inventions that change lives! The key is to follow the steps you already know: **Empathize, Define, Ideate, Prototype,** and **Test.**



Case Study 1: The Mobile Medical Clinic 🏥

Imagine a small town far away from any hospital. If someone gets sick, they have to travel for hours to see a doctor! That's a huge problem, and it affects everyone in the town.

Designers used Design Thinking to solve this problem by creating a **Mobile Medical Clinic**—a hospital on wheels!

Here is how the Design Thinking steps helped create this life-saving solution:

Step 1: Empathize (The Need) 🧑‍🤝‍🧑

Designers didn't just guess what the people needed. They spent time in the distant communities!

- They **listened** to people talk about how long it takes to travel to a clinic.
- They **observed** that the roads were often bumpy and muddy.
- They **understood** that the clinic needed to carry many different supplies, not just one type of medicine.

Step 2: Define (The Focus) 🎯

Based on their research, the designers wrote a clear problem statement.

- **Problem Statement:** "How might we design a reliable and fully-equipped medical facility that can reach remote communities safely and quickly, even on rough roads?"

Step 3: Ideate (The Ideas) 💡

The team brainstormed many ideas, focusing on transportation and equipment.

- *Silly Idea:* Build a flying drone hospital! (Too expensive and hard to control.)
- *Practical Idea:* Use a large bus or truck (like an 18-wheeler) and convert it into a clinic.
- *Key Feature:* The vehicle must be very strong and have a great suspension system to handle the bumpy roads.

Step 4: Prototype (The Model) 🧱

The team built a simplified version of the clinic.

- They used **computer-aided design (CAD)** software to create a 3D model of the truck's inside.
- They built a wooden frame inside a regular truck to **test the layout** of the examination room and how the equipment would be stored during travel.

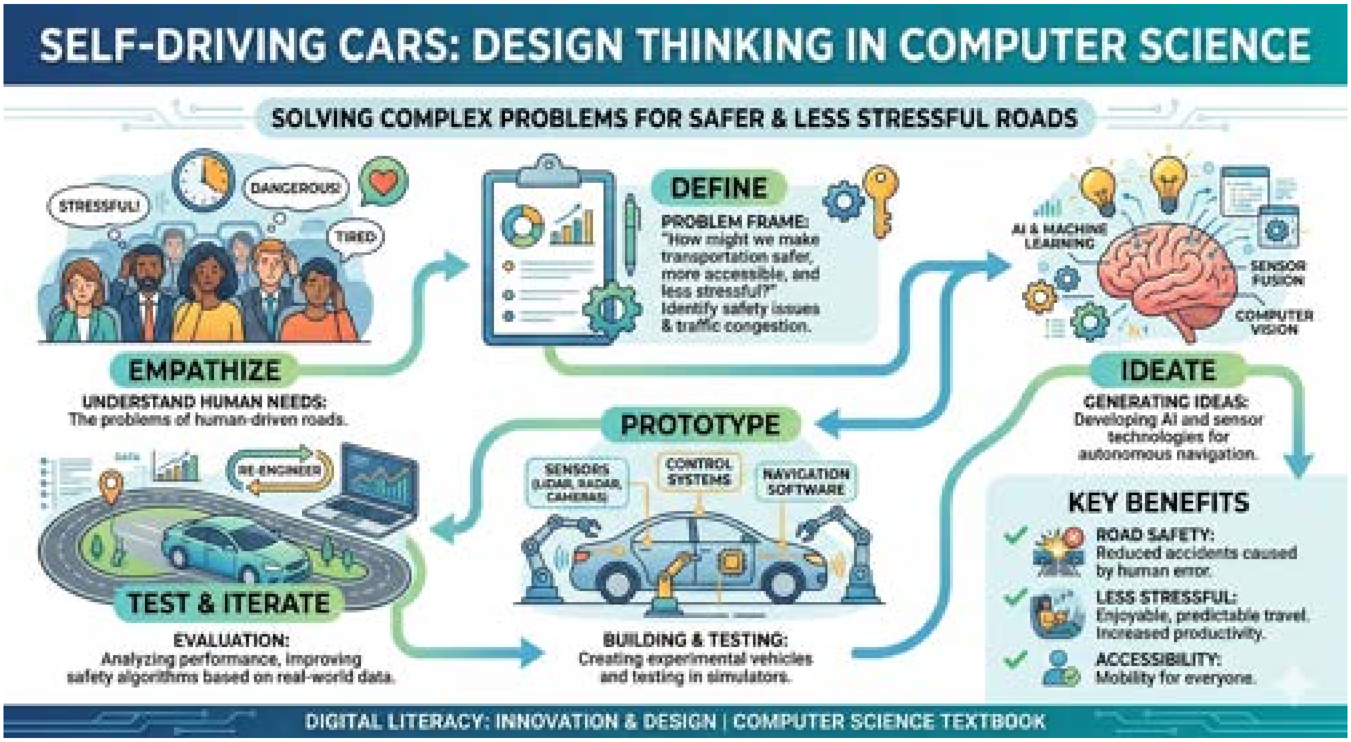
Step 5: Test (The Feedback) 💬

The team drove the prototype on rough roads and showed it to community leaders and doctors.

- **Feedback:** Doctors said the shelves were too high and things would fall off during a bumpy ride.
- **Result:** The designers went back to the **Prototype** step to lower the shelves and add special locks. This iterative process led to a final, successful Mobile Medical Clinic!

Case Study 2: The Self-Driving Car 🚗💻

Self-driving cars are an amazing example of Design Thinking in computer science. They solve the complex problem of making our roads safer and less stressful!



Step 1: Empathize (Road Safety) 🧑‍🤝‍🧑

The designers looked at the real-world problems caused by human drivers.

- They **gathered data** on accidents caused by tired or distracted drivers.
- They **studied** what makes people feel stressed while driving, like traffic jams.
- **Need:** People need safer, more relaxing transportation.

Step 2: Define (The Core Challenge) 🎯

The core challenge of self-driving cars is not just moving, but making complex, split-second decisions!

- **Problem Statement:** "How might we design an artificial intelligence (AI) system that can safely see, understand, and react to changing road conditions (like sudden rain or a bicycle) faster and more reliably than a human driver?"

Step 3: Ideate (The Technology) 💡

This step required thinking of many technologies working together.

- *Idea:* Use video cameras (like the ones on your phone).
- *Idea:* Use **LiDAR** (a laser technology) to "see" in 3D.
- *Idea:* Develop complex **algorithms** (computer science instructions) to predict where a pedestrian will walk next.

Step 4: Prototype (The Trial Run) 🧱

The prototype involved building and testing the AI software many times.

- They started by programming the car to only drive straight on an empty street.
- They created **virtual reality simulations** of complex city driving before testing on real roads.
- **Early Prototype:** A car that was limited to 10 mph and had a safety driver ready to take over at any moment.

Step 5: Test (Massive Data Collection) 💬

Self-driving car companies test their prototypes by driving millions of miles!

- Every time the car makes a wrong decision, the engineers learn from the data and **go back to the Ideate step** to create a better algorithm.
- The system is constantly being **iterated** (improved) based on new challenges, like driving in a snowstorm or navigating an unusual traffic circle.

The self-driving car shows how Design Thinking helps engineers manage many technologies (AI, sensors, software) to solve a truly complex, worldwide problem: transportation safety and efficiency. 🎉

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Chapter 4: Design Thinking: Solving Complex Problems

Copy Grade 4 CS Chapter 4 Lesson Plans

View Grade 4 CS Chapter 4 Podcast

View Grade 4 CS Chapter 4 Video

View Grade 4 CS Chapter 4 Infographic

Copy Grade 4 CS Chapter 4 Slides

Copy Grade 4 CS Chapter 4 Self Guided Learning Worksheet

Copy Grade 4 CS Chapter 4 Scratch Worksheet

Copy Grade 4 CS Chapter 4 micro:bit Worksheet

Copy Grade 4 CS Chapter 4 Unplugged Worksheet

Open Form: Grade 4 CS Chapter 4 Quiz

Design Considerations: Availability of Materials and Cost 💰

When you design and build things, whether it's a computer program, a new type of robot, or even a simple birdhouse, there are two really important things you must always think about: the materials you need and how much they cost. These two factors can change your whole design!

What is a Design? 💻

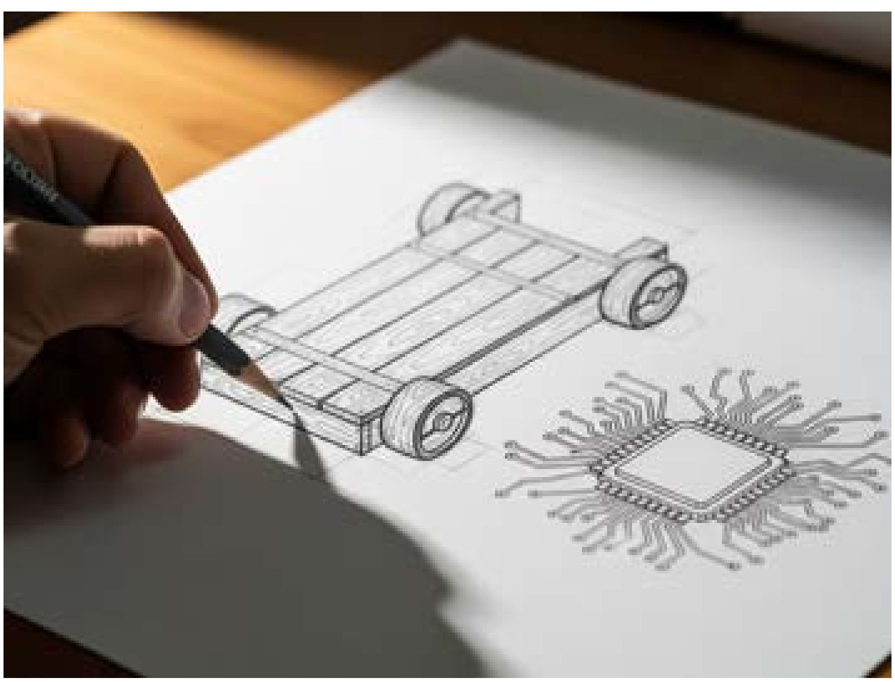
A **design** is a detailed plan or drawing that shows how something is going to be built or created. Think of it like a recipe 🍳 for making something. In computer science and engineering, the design phase is when you solve problems on paper (or on a screen!) before you start building.

Imagine you are designing a new school desk. Your design needs to consider:

- **Function:** Does it hold books and a laptop?
- **Safety:** Are there sharp edges?
- **Comfort:** Is it the right height for a Grade 6 student?
- **Materials:** What will it be made of (wood, metal, plastic)?
- **Cost:** Can the school afford to buy hundreds of them?

🧱 The Importance of Material Availability

Availability means how easy it is to get a material when you need it. If a material is hard to find, it can cause big problems for your design and production.



Case Study: Wood vs. Rare Earth Metals

Let's look at two examples:

1. **Building a wooden robot chassis:** If you design a robot body (chassis) out of pine wood, this is usually easy to find at a local lumber yard. Pine is a very **available** material. This means you can build many robots quickly!
2. **Building a computer chip:** Computer chips use special materials called **rare earth metals**. These metals are only mined in a few places in the world and can be difficult to process. They are **less available** than pine wood. If the supply of these metals suddenly stops, production of new computers stops too! This forces designers to look for alternative materials or redesign the chip entirely.

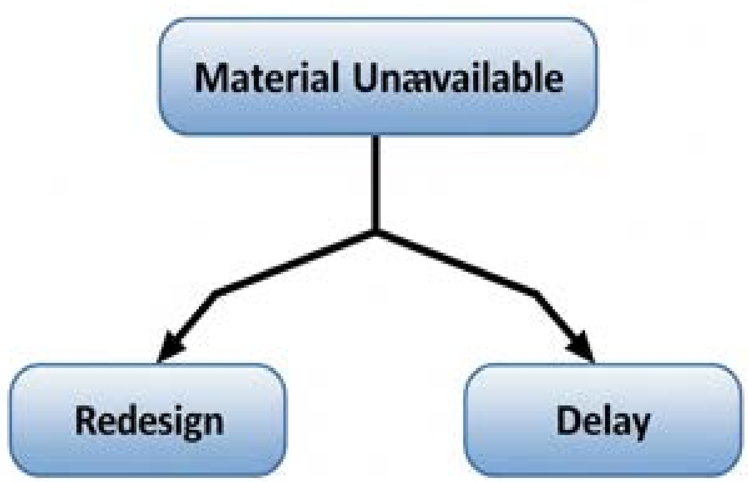
The availability of materials directly impacts how reliable your production process is.

Material	Availability	Impact on Design
Pine Wood	High	Easy to source, reliable production.
Rare Earth Metals	Low/Fluctuating	Production can stop; designers need backups.

If a material you planned to use is suddenly unavailable, you have two main options:

1. **Redesign:** Change the structure or function so it can be built with a different, available material.
2. **Delay:** Wait until the material becomes available (which might cost you time and money).

This is a key part of problem-solving in design! 💡



💸 The Role of Cost in Design

Cost is how much money you need to spend to get the materials. Every design has a budget, and you must stay within it. Using expensive materials means you can build fewer items, or you might have to sell your finished product for a very high price.

Total Cost Consideration

When we talk about cost, we don't just mean the price tag of the material itself. A designer must think about the **total cost**, which includes:

- **Material Price:** The cost to buy the raw material (e.g., a sheet of aluminum).
- **Processing Cost:** The cost to shape, cut, or prepare the material (e.g., 3D printing time, cutting wood).
- **Shipping Cost:** The cost to transport the material from where it's made to where you will use it.
- **Labor Cost:** The cost to pay the people who will assemble the design.

Cost and Material Choice

Sometimes, a material might seem cheap at first, but it increases the total cost later.

Example: Using very thin, cheap plastic for a phone case might save money on the material, but it might break easily. If it breaks quickly, the company will have to replace it (a warranty cost), which actually makes the design more expensive in the long run! A slightly more expensive, durable plastic might be the better choice overall.

Good designers always try to find the "sweet spot" where a material is:

1. **Available**
2. **Affordable**
3. **Suitable** (It does the job well!)

This continuous balancing act between these three factors is why design is a challenging and exciting field! 🎉

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Chapter 5: Design Considerations: Materials and Cost

Copy Grade 4 CS Chapter 5 Lesson Plans

View Grade 4 CS Chapter 5 Podcast

View Grade 4 CS Chapter 5 Infographic

View Grade 4 CS Chapter 5 Video

Copy Grade 4 CS Chapter 5 Slides

Copy Grade 4 CS Chapter 5 Self Guided Learning Worksheet

Copy Grade 4 CS Chapter 5 Scratch Worksheet

Copy Grade 4 CS Chapter 5 micro:bit Worksheet

Copy Grade 4 CS Chapter 5 Unplugged Worksheet

Open Form: Grade 4 CS Chapter 5 Quiz

Algorithms: Following Instructions

What is an Algorithm? 🤖

An algorithm is a set of precise, step-by-step instructions used to solve a problem or complete a task. Think of it like a recipe for a computer! Computers and other technology follow algorithms all the time to do things like find the best route on a map, sort your favorite songs, or even show you results when you search online.

Every algorithm must have a clear beginning and a clear end. If the steps are not clear, or if they are missing, the task won't be completed correctly.

Why are Algorithms Important?

Algorithms are the building blocks of all computer programs and apps. Without them, your tablet, phone, or video game console would not know what to do! Algorithms help us organize our steps and make sure the desired result is achieved every single time.

Think about making toast. If your instructions were: "Put bread in toaster. Wait a bit. Take toast out." This is not a good algorithm because "Wait a bit" is not precise enough. How long do you wait? The toast might burn!

A better algorithm for making toast would be:

1. Plug in the toaster.
2. Place one slice of bread into the toaster slot.
3. Push the lever down to start toasting.
4. Set the dial to level 3 (medium brown).
5. Wait for the toaster to pop up.
6. Carefully remove the toast.

This set of instructions is much clearer and will consistently produce medium-brown toast.

Algorithms in Real Life 🌍

You use algorithms every day without even realizing it! An algorithm is just an ordered sequence of steps. Here are a few examples:

- **Getting Ready for School:** The steps you take from waking up to leaving your house (wake up, brush teeth, get dressed, eat breakfast, pack bag).
- **Washing Your Hands:** (Turn on water, wet hands, add soap, rub hands for 20 seconds, rinse hands, turn off water, dry hands).
- **Finding a Page in a Book:** (Open to the Table of Contents, find the chapter title, look at the page number, turn to that page).

Visualizing an Algorithm

Sometimes, it is easier to understand an algorithm by drawing it out like a diagram. We can use drawings or symbols to show the different parts of the sequence.

Key Features of a Good Algorithm

A good algorithm has three main features:

Feature	Description
Input	The information or ingredients you start with. 🍞
Process	The steps or instructions that are followed. ⚙️
Output	The final result or the finished product. 😋

Let's look at the toast example again:

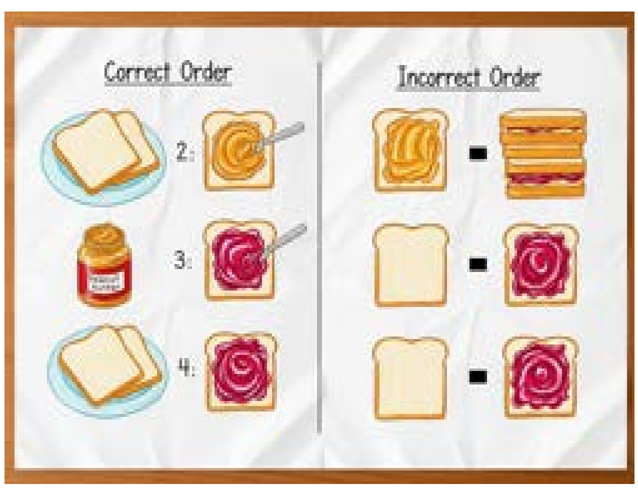
Feature	Toast Example
Input	A slice of bread, a toaster, electricity
Process	The 6 steps listed previously (plug in, place bread, push lever, etc.)
Output	A piece of medium-brown toast

When you write an algorithm for a computer, you must be extremely detailed. Computers can only follow exactly what you tell them! If you miss a step, the computer will either stop, crash, or give you the wrong answer.

Sequence of Instructions 🔢

The word **sequence** simply means that the instructions must be followed in a specific order. If you try to follow the steps out of order, the outcome will be different, or the task might fail.

Imagine trying to bake cookies. If you put the cookie dough in the oven **before** you mix the ingredients, you won't get a proper cookie! The sequence of instructions matters.



This is what an algorithm is all about: an ordered, logical sequence of instructions that consistently solves a problem. It is the language of computing! 💻

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Chapter 6: Algorithms: Following Instructions

Copy Gr 4 CS Chapter 6 Lesson Plans

View Grade 4 CS Chapter 6 Podcast

View Grade 4 CS Chapter 6 Infographic

View Grade 4 CS Chapter 6 Video

Copy Grade 4 CS Chapter 6 Slides

Copy Grade 4 CS Chapter 6 Self Guided Learning Worksheet

Copy Grade 4 CS Chapter 6 Scratch Worksheet

Copy Grade 4 CS Chapter 6 micro:bit Worksheet

Copy Grade 4 CS Chapter 6 Unplugged Worksheet

Open Form: Grade 4 CS Chapter 6 Quiz

Chapter Review: Design and Transformation

This chapter reviews the core idea that **Design involves processes that can transform ideas into artifacts that meet needs.** This outcome is central to understanding how technology and innovation shape the world around us.

💡 What is Design?

Design is more than just making something look pretty. It's a way of thinking and a systematic process for solving problems. When we say "design," we are talking about the careful planning and deliberate effort used to create something new.

- **Idea:** Design always starts with an idea. This could be a solution to a problem, a new way to do an old task, or just something cool you want to build.
- **Need:** Good design always addresses a need. Whether the need is for a faster way to travel, a safer way to store food, or a more engaging video game, the goal is to make life better or easier.
- **Process:** The process is the journey from the idea to the final product. It involves steps like researching, brainstorming, prototyping, and testing.

The Design Process: From Idea to Artifact 🛠️

The design process is a cycle that helps transform abstract ideas into tangible **artifacts**. An artifact is any object that is made by a human being, like a pencil, a computer program, a bridge, or a shoe.

1. Identify the Need

Before you can solve a problem, you must clearly define it.

Step	Description	Visual Example
Define the Problem	What is the challenge? Who has this need? What are the constraints (time, materials, cost)?	A simple drawing of a person struggling to carry too many books, with a thought bubble that says "My bag is too small!"

2. Brainstorm and Imagine Solutions

This is the creative part! You generate as many ideas as possible. No idea is too silly at this stage.

- **Sketching:** Quickly drawing out different possibilities.
- **Mind Mapping:** Connecting related ideas to see new solutions.

3. Plan and Model

Once you have a strong idea, you need to plan how it will be built. This often involves creating a model or a blueprint.

- **Blueprints:** Detailed drawings that show the size, shape, and parts of the artifact.
- **Digital Models:** Using computer software to create a 3D version of the design before building it physically. This is where computer science (CS) skills are often used! 💻

Concept	Explanation
Input	The information or energy put into a system (e.g., button press).
Process	The steps the system takes to handle the input (e.g., code runs).
Output	The result or artifact created (e.g., light turns on).

4. Create the Artifact (Prototype)

A prototype is the first version of your design. It's meant to be tested, not necessarily the final product.

- **Building:** Using materials like wood, plastic, or code to make a physical or digital model.
- **Testing:** Does the prototype actually meet the need you identified in Step 1?

5. Test, Evaluate, and Improve 🔄

Testing almost always reveals problems! That's okay—it's part of the process. Good designers use this feedback to make their artifact better.

- **Evaluation:** Asking questions like: "Did it solve the problem?" "Is it easy to use?" "Is it durable?"
- **Iteration:** The process of repeating steps 2, 3, and 4 to refine the design. Design is not a straight line; it's a loop!

Computer Science and Design 🤖

Computer Science (CS) is a huge part of modern design. Creating digital artifacts—like apps, websites, or video games—is a design process powered by code.

- **User Interface (UI) Design:** Designing the buttons, screens, and visuals of an app to make it look good.
- **User Experience (UX) Design:** Designing how an app feels to use, making sure the process is smooth and logical.

The most powerful transformation of an idea happens when a designer uses code to bring a digital artifact to life.

CS Concept	Role in Design
Algorithms	The set of steps (like a recipe) that the artifact follows to work.
Data	The information the artifact uses or generates (e.g., scores in a game).
Logic	The rules that determine how the artifact behaves (if X, then Y).

Google Drive Resources:

1. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
2. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
3. Google Forms will open the responder page. You can print this page to PDF.
   Tip! Google Gemini in Forms will automatically turn this PDF into a proper Form.

Copy Grade 4 CS Conclusion: Design and Transformation Review

Copy Grade 4 CS Conclusion Lesson Plan (Final Project)

Copy Grade 4 CS Final Project Instructions

Copy Grade 4 CS Conclusion: Peer Feedback Form for Design Iteration

Copy Grade 4 CS Conclusion Rubric

Copy Grade 4 CS Conclusion: Design Thinking Project Portfolio Worksheet

Copy Grade 4 CS Conclusion: Design Thinking Rubric For Students