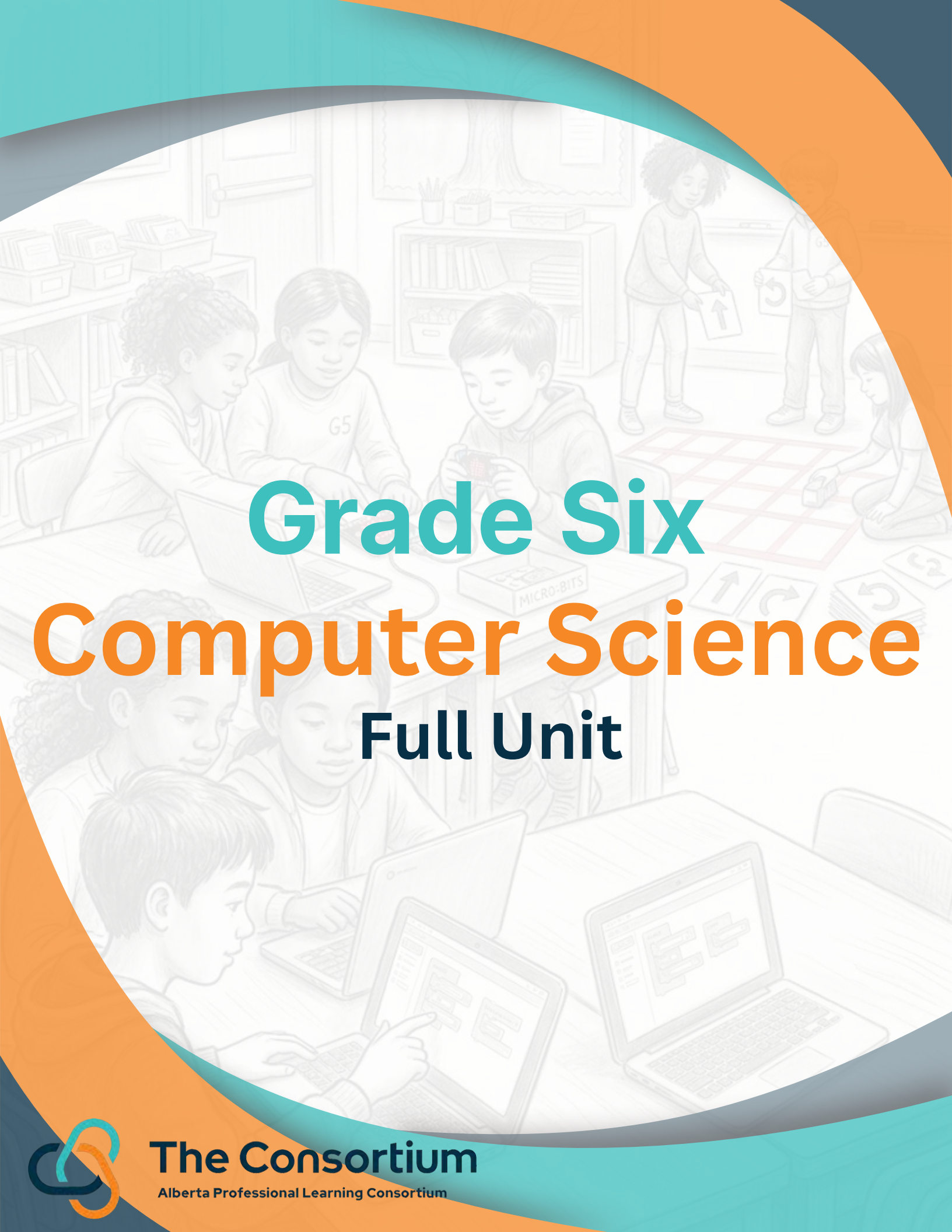


Grade Six Computer Science

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 6 CS All Lesson Plans

Open Form: Grade 6 Computer Science Final Exam: Abstraction, Code, and Impact

Copy Grade 6 CS Full Text Book

Copy Grade 6 Computer Science

Introduction: Welcome to the Amazing World of Computer Science! 🚀

Have you ever wondered how your favorite apps work? Or how a video game knows exactly what to do when you press a button? That's all Computer Science! This chapter is your first step into understanding the magical and logical world of computers, coding, and technology. Get ready to explore how we can use awesome ideas to solve problems and create cool things!

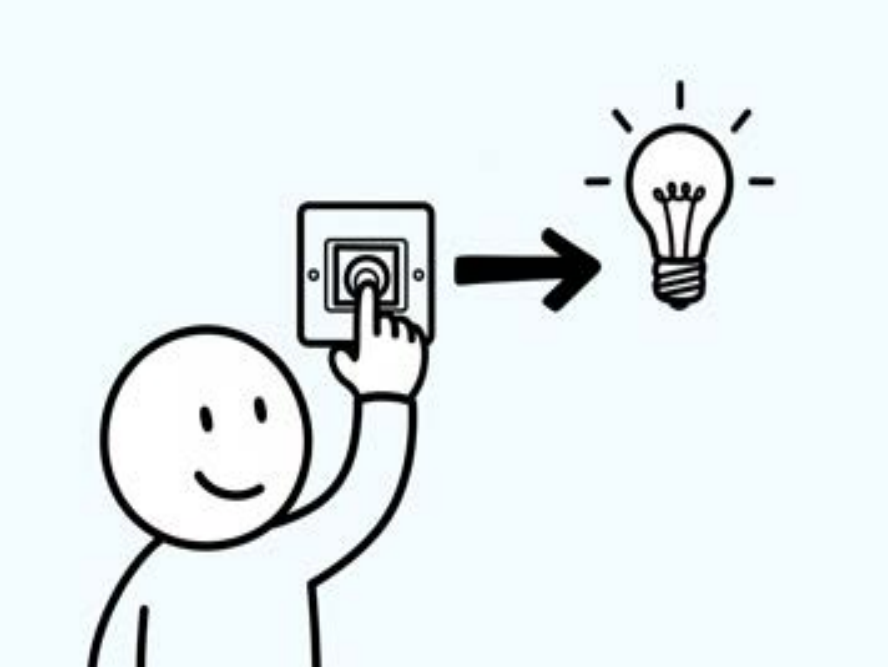
💡 What is Abstraction? Simplifying the Complex

Imagine trying to explain everything about a car: the engine, the battery, the brakes, the fuel lines... it's a lot! But when you drive a car, what do you actually need to worry about? The steering wheel, the pedals, and the light switches!

Abstraction is a fancy word for something simple: it’s a **simplified version of something complex**. 🤯

It’s about focusing on what is important and ignoring the stuff you don't need right now. Think of it like this:

The Complex Thing	The Simple Abstraction	Why it's easier
The hundreds of wires and electrical components behind the wall	A light switch 💡	You just flip it to turn on the light!
The thousands of engine parts and mechanics in a car	A steering wheel 🚗	You just turn it to change direction!
The millions of lines of code that make a weather app	A simple icon and the temperature display ☀️	You just tap the app to see the weather!



The Process of Abstraction

When we use abstraction in design and coding, we follow these steps:

- **Determining what details to keep and what to ignore:** Do we need to know the wire color? Probably not!
- **Removing unnecessary details:** Get rid of the clutter!
- **Identifying important information:** What is the one thing we *must* know?
- **Generalizing patterns:** If all cars have steering wheels, that's an important pattern!

Skill Check: Can you think of an app on your phone or tablet? What complex things does it hide with a simple button or screen? 🤔 (This is **identifying examples of abstractions encountered in daily life**!)

💻 Coding Structures: Telling the Computer What to Do

Coding is like giving a computer a recipe. You have to be very clear, and you use special "structures" to organize your instructions.

1. Sequences (Order Matters! 📏)

A **sequence structure** is just an ordered set of instructions. The computer does instruction 1, then instruction 2, then instruction 3, and so on.

Example Recipe:

1. Put on your shoes.
2. Walk to the door.
3. Open the door.
4. Step outside.

2. Conditionals (If This, Then That! 🚦)

A **conditional structure** tells the computer to do different things based on different situations. We often use "if-then-else" statements.

Situation	Action 1 (If True)	Action 2 (Else/If False)
IF the traffic light is RED	**THEN** STOP.	**ELSE** (If it is green or yellow) GO.

3. Loops (Do it Again! 🔄)

Loops tell the computer to repeat a set of instructions over and over.

Instruction Set	How many times?
Take a step.	Repeat 100 times to walk across the field.

We will use a visual block-based language to practice designing code with these structures! (This is **using a visual block-based language to design code that includes relevant design structures**.)

🌐 Computer Science in Our Lives: Making a Difference

Computational artifacts are the tools and programs we design, and they are everywhere! They are designed to address needs and wants in society. (This is **discussing the role of design and coding in society**!)

Area	Computational Artifact Example
Weather	Weather modeling apps help us predict storms ⛈️
Communication	Phones and text messages connect people 💬
Transportation	Automotive controls (like cruise control) make driving easier
Medicine	Apps and programs help with medical research and diagnosis

🤔 The Big Picture: Impacts of Technology

Computers, coding, and technology can have huge impacts—some good, and some we need to watch out for! These impacts can be:

- **Personal:** How technology affects you (e.g., spending less time outside).
- **Social:** How technology affects groups of people (e.g., connecting friends through social media).
- **Environmental:** How technology affects the planet (e.g., energy used by huge computer servers).
- **Economic:** How technology affects money and jobs (e.g., online shopping).

Sometimes, these impacts are **intentional** (we meant for it to happen), and sometimes they are **unintentional** (a surprise!).

Challenge: Think about a popular video game. What is a positive impact (e.g., fun, connecting with friends) and what might be a negative impact (e.g., distraction from homework)? (This is **predicting possible impacts of computers, coding, or technology** and **discussing how computers, coding, or technology have had impacts**.)

Google Drive Resources:

4. Docs, Sheets and Slides will open a Template for you to use to make your own copy.
5. mp3's (audio) and mp4's (video) will give you view only access. Make a copy!
6. 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.

View Grade 6 CS Introduction Video

View Grade 6 CS Introduction Podcast

View Grade 6 CS Introduction Infographic

Copy Grade 6 CS Introduction Slides

Copy Grade 6 CS Introduction Worksheet

Copy Grade 6 CS Introduction Lesson Plan

Copy Grade 6 CS Introduction: The World of Computer Science

Chapter 1: Information and Abstraction: Making Sense of the World

📚 What is Information?

Imagine a giant pile of Lego bricks scattered all over the floor. This giant, unorganized pile is like **data**. Data is a collection of raw facts, figures, numbers, or symbols. On its own, it can be overwhelming and not very useful.

Data Example	Description
23, 17, 30, 20, 18	A list of numbers that don't mean anything on their own.
Red, Blue, Green	A list of colors.
⛈️, ☀️, 🌧️	A collection of weather symbols.

When you take those scattered Lego bricks and follow the instructions to build a spaceship, you are **organizing** them. This organized and useful collection of data is called **information**.

Information is data that has been organized, processed, and structured to be more useful and meaningful. 🧠

For example, if you organize the numbers 23, 17, 30, 20, 18 and label them as the *number of students in Grade 6 classes (A, B, C, D, E)*, they now tell you something useful.

Organized Data (Information)	Description
Grade 6 Class A: 23 students	This is useful information: you know how many students are in Class A.
Average Temperature Last Week: 20°C	The data (temperatures each day) has been organized and processed into one meaningful number.

The key difference is that **data is raw**, and **information is useful**! 👍

🧩 Understanding Abstraction

Have you ever tried to explain a complicated board game to a friend? You probably didn't read every single rule on the box. Instead, you told them the main goal and the few most important steps to get started. You were creating an **abstraction**!

An **abstraction is a simplified version of something complex**. It focuses on the most important details and hides the complicated, unnecessary details.

Think of it like an ice cream cone. 🍦 You only need to worry about the delicious ice cream on top and the cone you hold. You don't need to know every complicated step the company took to make the ice cream and the cone. The cone is an abstraction of the full, complex manufacturing process.

Abstractions in Daily Life

Abstractions are all around us and make our daily lives much easier! They allow us to use complex technology without needing to be an expert in how they work.

1. Simple Controls on Appliances 📺

When you turn on your TV, you don't have to worry about the thousands of tiny wires, circuits, and computer chips inside. All you see is a simple remote control with buttons for **Volume**, **Channel**, and **Power**.

- **Complex Reality:** Electrical signals, digital processors, screen technology.
- **Abstraction:** The simple remote control buttons. 🕹️
- **Benefit:** You can watch your favorite show without being an engineer!

2. The Light Switch 💡

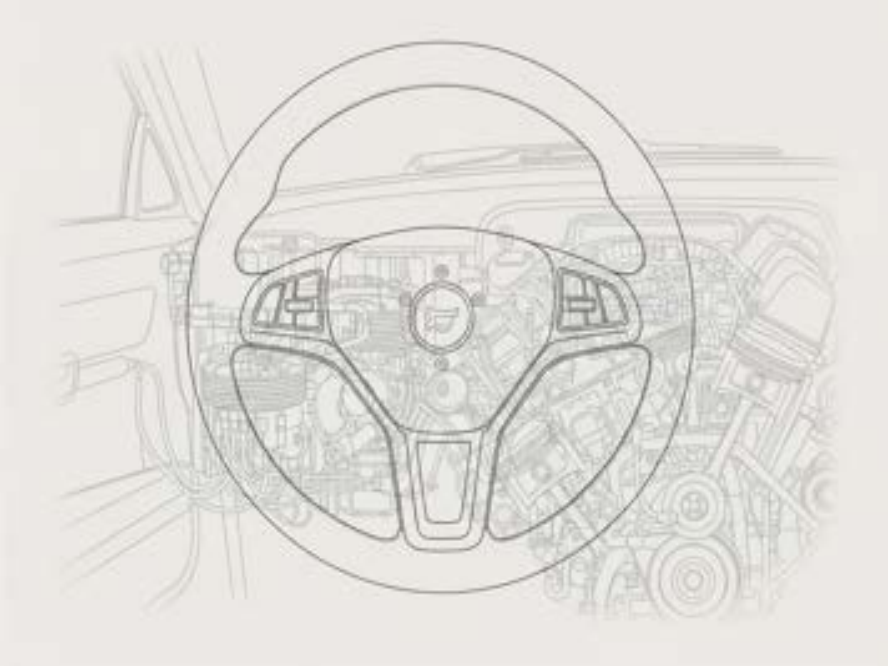
When you flip a light switch, you expect the light to turn on or off instantly.

- **Complex Reality:** Wires running through walls, an electrical circuit opening and closing, and power flowing from the grid.
- **Abstraction:** The simple "on/off" switch.
- **Benefit:** You can light up a room with a single finger movement. Imagine if you had to manually connect the wires every time!

3. Steering Wheels 🚗

Driving a car involves a huge amount of complex engineering, including gears, axles, and hydraulics.

- **Complex Reality:** Detailed mechanics connecting the wheels to the motor and steering column.
- **Abstraction:** The simple round steering wheel.
- **Benefit:** You can turn the car with a simple rotation, without needing to understand the engine.



4. Apps (Applications) 📱

When you use a phone app to check the weather or send a text, you are seeing a massive abstraction.

- **Complex Reality:** Computer code (lines of instructions), data storage on a server, and networking cables running across the world.
- **Abstraction:** The colorful icon on your screen and the simple user interface.
- **Benefit:** You can connect with friends and access information instantly just by tapping an icon.

Abstractions help us manage complexity by hiding what we don't need to see and showing only what is necessary and useful. They turn complicated processes into simple tools. What other abstractions do you use every day? 🤔

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 Information & Abstraction: Student Worksheet

View Grade 6 CS Chapter 1 Infographic

View Grade 6 CS Chapter 1 Podcast

View Grade 6 CS Chapter 1 Video

Copy Grade 6 CS Chapter 1 Slides

Copy Grade 6 CS Chapter 1 Lesson Plans

Copy Gr 6 CS Chapter 1 Unplugged Activity Worksheet

Copy Gr 6 CS Chapter 1 Scratch Activity Worksheet

Copy Gr 6 CS Chapter 1 micro:bit Activity Worksheet

Copy Grade 6 CS Chapter 1: Understanding Information and Abstraction

Chapter 2: Understanding Abstraction: The Power of Simplifying

Welcome to the exciting world of **Abstraction**! 🌍 Abstraction is a big, important idea in computer science and everyday life. Simply put, abstraction is the process of hiding complicated details and showing only the essential information. It's about making things simpler so we can focus on what really matters. Think of it like taking a giant, messy pile of toys and putting away all the ones you don't need, leaving only the few you want to play with right now. 🧸

1. Determining What Details to Keep and What to Ignore 🤔

The first step in abstraction is deciding what information is **important** and what can be **ignored**. This is a critical thinking skill that helps us solve problems more easily.

Imagine you are describing your favorite dog to a friend who lives far away.

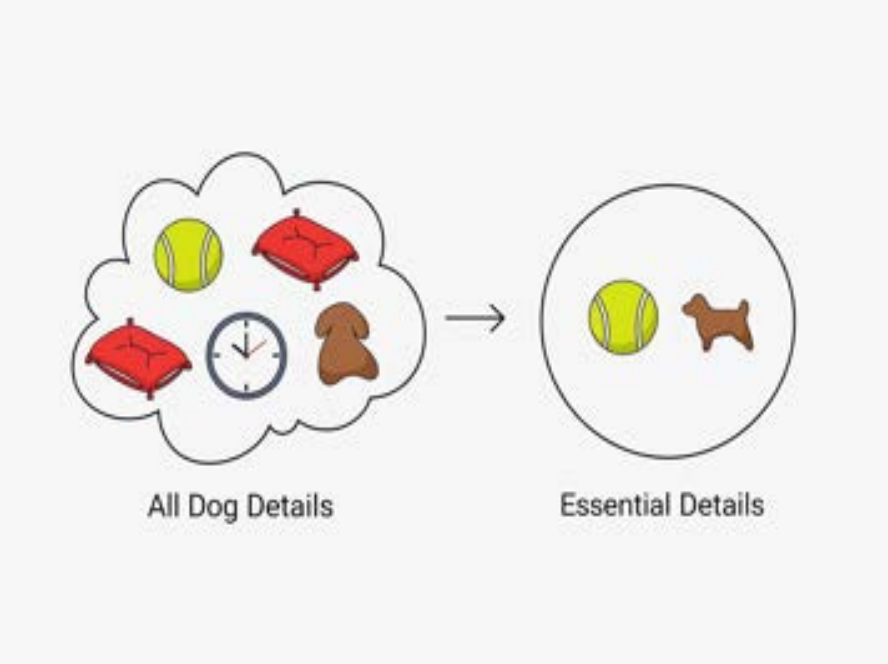
Here are some details about the dog:

- It has brown fur.
- It is a Golden Retriever.
- It weighs 65 pounds.
- It is sitting on a small, red cushion right now.
- It loves to chase tennis balls.
- It just ate a treat that took 12 seconds to chew.

If you want your friend to understand what kind of dog you have and what it likes to do, which details are important to keep, and which ones can be ignored?

Detail	Keep (Important)	Ignore (Unnecessary)
Brown fur	Yes	
Golden Retriever	Yes	
Weighs 65 pounds	Yes	
Sitting on a red cushion		Yes
Loves to chase tennis balls	Yes	
Just ate a treat in 12 seconds		Yes

You would keep the details that describe the **identity** and **behavior** of the dog, and you would ignore temporary details like what it's sitting on or how long it took to eat a treat.



2. Removing Unnecessary Details ✂️

Once you have decided which details to ignore, the next step is actually removing them! This process creates a simpler, cleaner, and more useful model of the thing you are describing.

Think about a map. 🗺️ A map is a perfect example of abstraction.

What details does a map **remove** to be useful?

- Individual trees 🌳
- Every car parked on the street 🚗
- The colors of every house 🏠
- People walking around 🚶

Why are these details unnecessary? Because a map's main job is to show you the **roads**, **landmarks**, and **routes** between locations. Adding every single tree or car would make the map so cluttered and complicated that you couldn't find your way! By removing these unnecessary details, the map becomes an abstract representation—a tool that is much easier to use.

3. Identifying Important Information ⭐

Identifying important information is the core of abstraction. It means finding the most valuable, unchanging, or relevant facts that define something. This information forms the basis of your simplified model.

Let's use the example of an electronic clock. ⏰

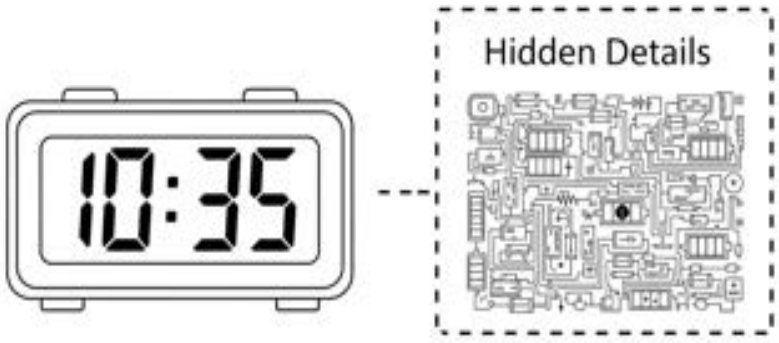
What is the single, most important piece of information that a clock provides?

The important information is the **time** (hours, minutes, and seconds).

What are some less important or hidden details?

- The complex wiring inside the plastic case.
- The tiny computer chip that runs the clock.
- The exact type of battery used.

When you look at an abstract clock face, you only see the important information: the numbers. All the complicated, technical information about *how* the clock works is hidden away. This is how abstraction works in technology! You don't need to be an engineer to read the time.



4. Generalizing Patterns 🔄

Generalization is the process of taking specific examples and finding a **common pattern** or rule that applies to all of them. This is how we create general concepts that save us time and energy.

Imagine you see the following sequence of instructions:

1. Feed the *cat* at 7:00 AM.
2. Feed the *dog* at 7:00 AM.
3. Feed the *fish* at 7:00 AM.

Instead of writing three separate instructions, you can **generalize** the pattern.

The pattern is: **All pets get fed at 7:00 AM.**

The generalized, abstract rule is:

"Feed **all pets** at 7:00 AM." 

This generalized rule is much simpler to remember and apply. In computer programming, generalizing patterns is essential for writing efficient code. Instead of writing the same code three times, a programmer writes one general rule that applies to all similar items.

Generalizing helps us create **templates** and **categories**. When you learn that a "chair" is something you sit on, you have generalized. You can then recognize a tiny wooden stool and a big beanbag as both being "chairs," even though they look very different. You have kept the important detail ("something to sit on") and ignored the unnecessary details (size, color, material). That's the power of abstraction! 💪

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 6 CS Chapter 2 Lesson Plans

View Grade 6 CS Chapter 2 Video

View Grade 6 CS Chapter 2 Podcast

View Grade 6 CS Chapter 2 Infographic

Copy Grade 6 CS Chapter 2 Slides

Copy Grade 6 CS Chapter 2 Self Guided Learning Worksheet

Copy Grade 6 CS Chapter 2 Scratch Activity Worksheet

Copy Grade 6 CS Chapter 2 Unplugged Activity Worksheet

Copy Grade 6 CS Chapter 2 micro:bit Activity Worksheet

Copy Grade 6 CS Chapter 2: The Power of Simplifying

Chapter 3: Computational Artifacts

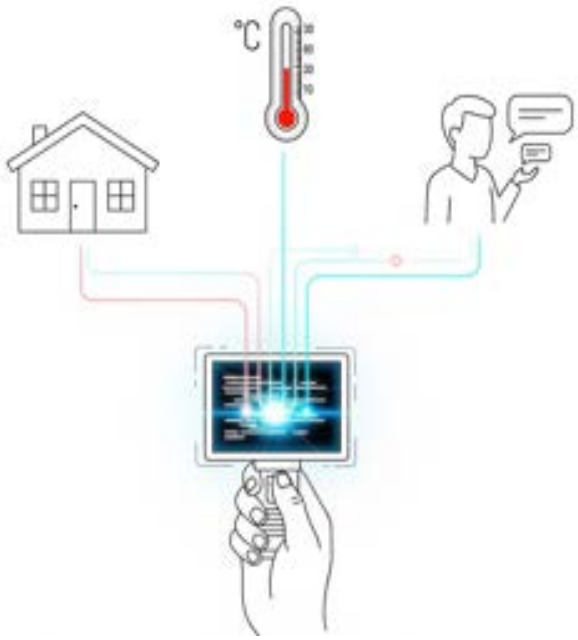
Welcome to a fascinating topic! In this chapter, we will explore computational artifacts.

What is a Computational Artifact? 🤔

The word "artifact" simply means something made by a human. A computational artifact is something created using a computer or code that helps us do something. Think of it as a tool made of logic, data, and instructions.

Component	Description	Example
Code	The instructions that tell the computer what to do.	Programming languages like Python or Scratch.
Data	The information the code uses or creates.	Temperature readings, images, or recorded voices.
Logic	The "rules" that guide how the code uses the data.	If it's cold, then turn on the heater.

Computational artifacts are designed to solve problems, make tasks easier, or satisfy a "want" in society.



Addressing Societal Needs and Wants

Society is just another word for all of us living and working together. We all have needs (things we must have, like safety or health) and wants (things that make life better, like entertainment or fast communication). Computational artifacts are powerful because they can address both!

1. Weather Modeling 🌦️☀️

One of the oldest needs of society is knowing about the weather. Farmers need it for crops, sailors need it for safety, and we all need it to decide what to wear!

Weather modeling uses huge amounts of data, like temperature, air pressure, and wind speed, and runs it through complex computer programs.

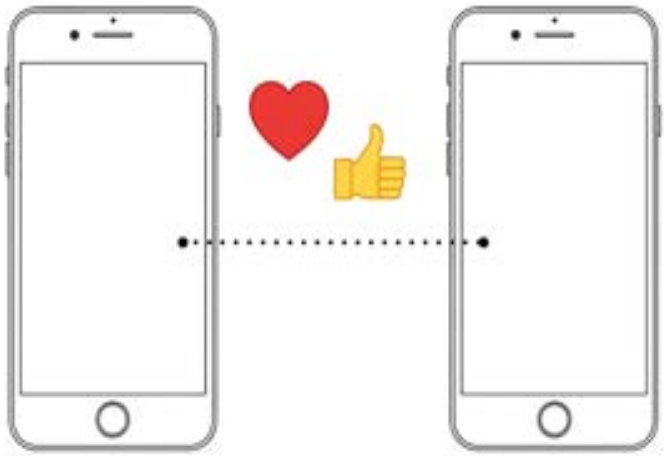
- **Need addressed:** Safety and planning (protecting property and lives).
- **How it works:** The code simulates, or imitates, how the atmosphere behaves. This simulation is the computational artifact. It helps us predict if a big storm is coming.

2. Communication 💬📧

Think about how you talk to friends or family far away. Computational artifacts have completely changed communication.

Apps like messaging services, video calls, and social media are all computational artifacts.

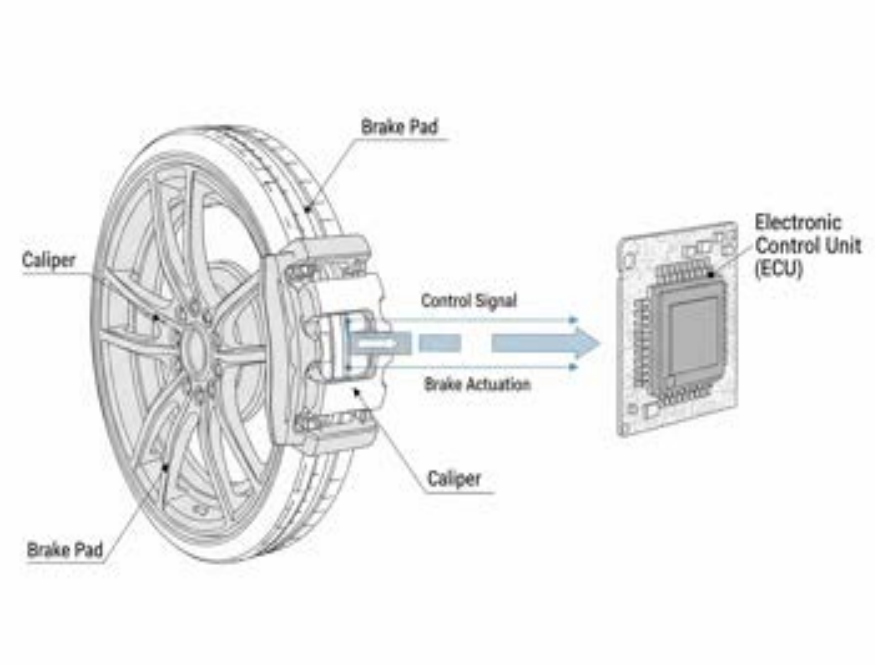
- **Want addressed:** The desire for instant connection and sharing information.
- **How it works:** When you send a message, the app (the artifact) takes your text or video, breaks it into tiny pieces of data, sends it across the internet using logic, and then puts it back together on the receiver's device.



3. Automotive Controls 🚗

Modern cars are full of tiny computers! These computers manage everything from the engine's fuel efficiency to your safety. These systems are powerful computational artifacts.

- **Need addressed:** Safety, efficiency, and transportation.
- **Example:** Anti-lock Braking Systems (ABS).
- **How it works:** A sensor (part of the artifact) detects if a wheel is about to lock up during braking. The computational artifact uses logic to rapidly pulse the brakes instead of locking them, which prevents the car from skidding.



4. Medical Research 🔬💊

Computational artifacts are crucial for improving our health and finding cures for diseases.

- **Need addressed:** Health, wellness, and curing diseases.
- **How it works:** Scientists use powerful computers to create models of diseases or human bodies. This lets them test thousands of medicines virtually, much faster and safer than testing in the real world. This virtual testing model is a computational artifact. It helps speed up the development of new treatments.

5. Apps 📱

An "app" is short for application, and nearly all apps are examples of computational artifacts designed to satisfy a specific need or want.

Apps can be simple, like a stopwatch, or complex, like an education app that tracks your learning progress.

Type of App	Need/Want Addressed
Navigation Apps	Need for direction and efficient travel.
Educational Apps	Want for easier and engaging learning.
Banking Apps	Need for managing money safely and conveniently.

Every time you use an app, you are interacting with a computational artifact designed by someone to solve a problem for you! That is the power of design and code! 💪

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 Graded 6 CS Chapter 3 Lesson Plans

View Graded 6 CS Chapter 3 Infographic

View Graded 6 CS Chapter 3 Podcast

View Grade 6 CS Chapter 3 Video

Copy Grade 6 CS Chapter 3 Slides

Copy Grade 6 CS Chapter 3 Scratch Activity Worksheet

Copy Grade 6 CS Chapter 3 micro:bit Activity Worksheet

Copy Grade 6 CS Chapter 3 Unplugged Activity Worksheet

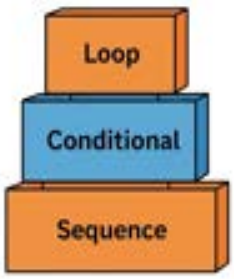
Copy Grade 6 CS Chapter 3 Self Guided Learning Worksheet

Copy Grade 6 CS Chapter 3: Computational Artifacts

Chapter 4: Introduction to Coding Structures 🤖

Welcome to the exciting world of coding! Think of coding as giving a set of instructions to a computer, just like following a recipe or a treasure map. To make sure the computer understands exactly what to do, we use special arrangements of instructions called **structures**.

These structures are the building blocks of every computer program and help us organize our code so it works perfectly. In this lesson, we will explore three of the most important structures: **sequences**, **conditionals**, and **loops**. Get ready to learn how to command a computer!



---

1. Sequences: The Ordered Path ➡️

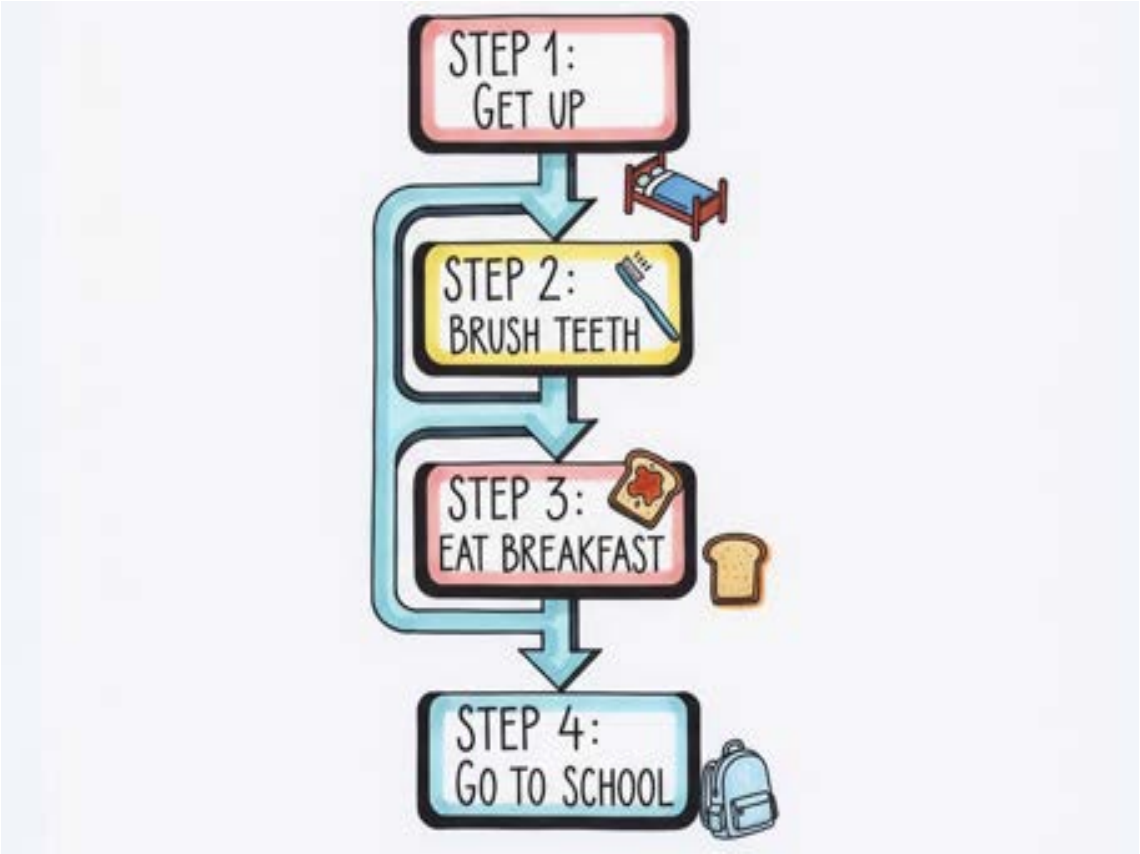
A **sequence** is the simplest and most common structure in coding. It is just an **ordered set of instructions** that the computer follows one after the other, from top to bottom.

Imagine you are baking a cake 🎂. You must follow the steps in the correct order, or the cake won't turn out right! The computer is the same—it follows the sequence in the exact order you write it.

How Sequences Work

Step	Instruction	Action
1	`Turn on the oven.`	The computer performs the first instruction.
2	`Mix flour and sugar.`	The computer performs the second instruction.
3	`Pour mixture into pan.`	The computer performs the third instruction.
4	`Bake for 30 minutes.`	The computer performs the final instruction.

Every line of code you write is part of a sequence, ensuring the instructions are always carried out in the intended order.



---

2. Conditionals: Making Decisions 🤔

Sometimes, a computer needs to make a decision and choose a different path based on a situation. This is where a **conditional structure** comes in handy!

Conditional structures are statements that tell computers to complete different actions based on different situations (or **conditions**).

Think about deciding what to wear outside 👕. The condition is the weather.

- **IF** it is raining, **THEN** wear a rain jacket.
- **ELSE** (if it is NOT raining), **THEN** wear a light sweater.

The IF-THEN-ELSE Statement

The most common conditional structure is the **IF-THEN-ELSE** statement.

Part	What it means	Example
IF	The computer checks a condition (is it true or false?).	**IF** `score` is greater than 100
THEN	If the condition is TRUE, the computer does *this* action.	**THEN** `print "You win!"`
ELSE	If the condition is FALSE, the computer does *this* alternative action.	**ELSE** `print "Keep trying!"`

A Simple Example

Imagine a video game where a player needs a key to open a door.

```
None
IF player_has_key == TRUE
  THEN open_door
ELSE
  display_message("Door is locked!")
```

This code tells the computer: Check if the player has the key. If yes, open the door. If no, show the "Door is locked!" message. This allows the program to react differently to different situations.

---

3. Loops: Repeating Actions 🔄

What if you need the computer to do the same thing many times? You could write the instruction over and over, but that would be a lot of work! Instead, we use a **loop structure**.

A **loop** is a structure that tells the computer to **repeat a set of instructions** until a specific condition is met or a certain number of times. Loops save time and make your code much shorter!

Types of Loops

There are a few ways to create a loop, but they all involve repetition:

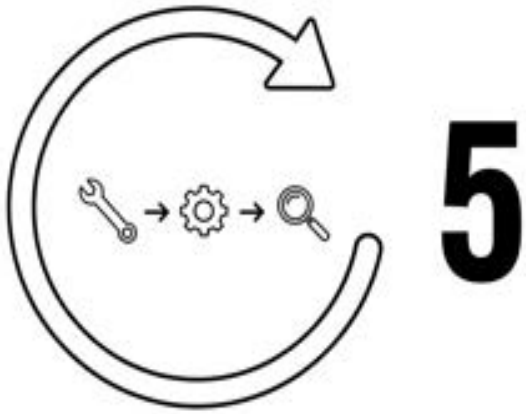
1. **Counted Loops (FOR Loop):** Repeat an action a specific number of times.
2. **Conditional Loops (WHILE Loop):** Repeat an action *as long as* a condition is true.

Example: Counted Loop (FOR)

If you want a robot to take 5 steps forward:

```
None
FOR 5 times
  take_one_step_forward
```

The computer will perform the `take_one_step_forward` instruction exactly 5 times.



Example: Conditional Loop (WHILE)

If you want a character to keep jumping until they reach the top of a wall:

```
None
WHILE character_height < wall_height
  jump_up
```

The computer will perform the `jump_up` instruction repeatedly, checking the condition (`character_height < wall_height`) before each jump. The loop stops as soon as the character's height is no longer less than the wall's height.

Loops are powerful tools for making computers do repetitive tasks quickly and efficiently! 🚀

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 6 CS Chapter 4 Lesson Plans

View Grade 6 CS Chapter 4 Infographic

View Grade 6 CS Chapter 4 Video

View Grade 6 CS Chapter 4 Podcast

Copy Grade 6 CS Chapter 4 Slides

Copy Grade 6 CS Chapter 4 Self Guided Learning Worksheet

Copy Grade 6 CS Chapter 4 Scratch Activity Worksheet

Copy Grade 6 CS Chapter 4 micro:bit Activity Worksheet

Copy Grade 6 CS Chapter 4 Unplugged Activity Worksheet

Copy Grade 6 CS Chapter 4: Coding Structures for Grade 6

Chapter 5: The Impacts of Technology

Welcome to a new chapter! We live in a world surrounded by technology. From the phones in our pockets to the computers in our classrooms, technology, and the code that powers it, shapes almost everything we do.

But have you ever stopped to think about all the ways this technology affects us and the world around us?

The use of computers, coding, and technology can have four main types of impacts:

1. **Personal** (How it affects you as an individual) 🧍
2. **Social** (How it affects groups of people and communities) 🏘️
3. **Environmental** (How it affects the natural world and the planet) 🌎
4. **Economic** (How it affects money, jobs, and businesses) 💰

We are going to explore each of these impacts in detail!

Personal Impacts: How Technology Affects You

Personal impacts are the changes and effects that technology has on an individual person. These impacts relate to how you think, feel, learn, and live your daily life.

Learning and Creativity

Technology has changed how we learn. Think about how you use a computer to find information for a school project—that's a personal impact!

- **Positive Impact:** Using educational apps or online tutorials can help you learn a difficult concept better. Technology can make learning more engaging and accessible. You can also use computers to create amazing art, music, or code your own games.
- **Negative Impact:** Spending too much time looking at a screen can cause eye strain or make it difficult to focus on other tasks.

Health and Well-being

Technology can affect your physical and mental health.

- **Positive Impact:** Smartwatches can track your steps and remind you to move, helping you stay healthy. You can easily connect with friends and family using video calls, which can improve your mood.
- **Negative Impact:** If you spend all your free time playing video games or scrolling on a phone, you might not get enough physical exercise or sleep. This can negatively affect your health.

Social Impacts: Technology and Our Communities

Social impacts are the changes that technology has on groups of people, communities, and how we interact with each other.

Connection and Communication

Technology makes it easier than ever to communicate with people across the street or across the world!

- **Positive Impact:** Social media platforms (when used safely) and messaging apps allow friends and family to stay connected, no matter the distance. This helps strengthen relationships and create communities.
- **Negative Impact:** Technology can sometimes lead to distraction during face-to-face conversations. If everyone is looking at their phone, it can make people feel less connected in real life. Also, things like cyberbullying or the spread of misinformation can be negative social impacts.

Access and Equality

Technology can help people access services and opportunities, but sometimes, not everyone has the same access.

- **Positive Impact:** Online government services or banking can make it easier for citizens to manage their lives. Technology helps doctors connect with patients who live far away (called telemedicine).
- **Negative Impact:** Sometimes, not every community has fast, reliable internet. This difference in access is called the "digital divide," and it means some people are left behind.

Environmental Impacts: Technology and Our Planet

Environmental impacts are the effects that computers, coding, and other technology have on the natural world, including the air, water, and land.

Resource Use and Waste

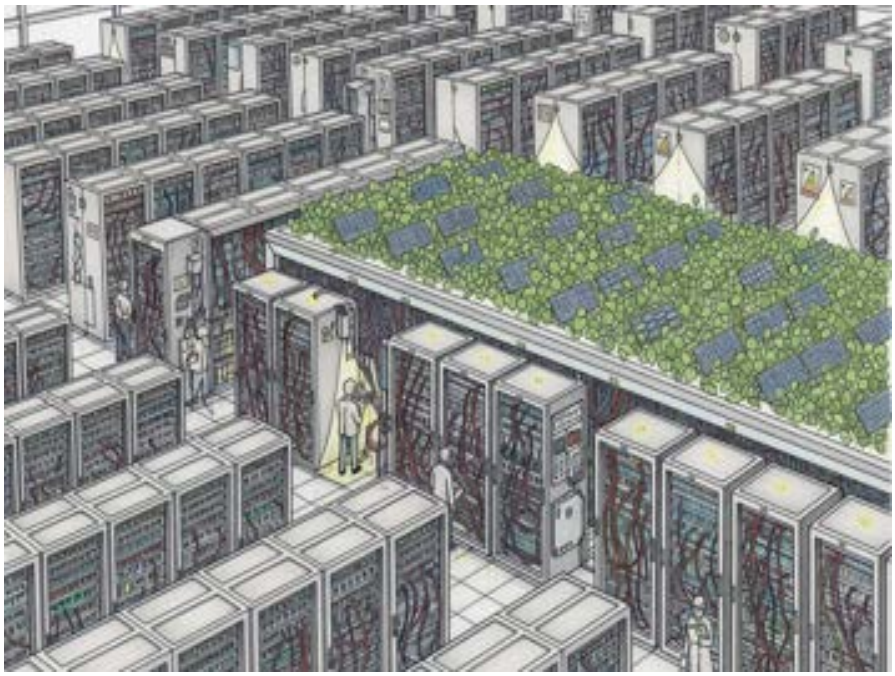
Every piece of technology, from a large computer server to a small battery, requires natural resources to be made.

- **Impact:** Manufacturing new devices uses materials like rare metals and plastics. When old devices are thrown away, they become **e-waste** (electronic waste), which can contain harmful chemicals that pollute the earth and water. 🗑️

Energy Consumption

Computers and data centers (huge buildings full of servers that power the internet) need a lot of electricity to run and stay cool.

- **Impact:** This massive energy use often relies on power plants that burn fossil fuels, which releases greenhouse gases and contributes to climate change. Coders and engineers are always looking for ways to make devices and data centers more energy-efficient!



Economic Impacts: Technology and Money

Economic impacts are the changes that technology creates in jobs, businesses, industries, and the flow of money.

New Jobs and Industries

Technology is constantly creating new types of work.

- **Positive Impact:** Coding and computer science have created millions of new, high-paying jobs, such as software developer, website designer, and cybersecurity expert. New companies are constantly being created based entirely on technology (like app companies).
- **Negative Impact:** As technology automates tasks, some existing jobs may disappear. For example, robots might replace people doing repetitive factory work. This is why learning new skills, like coding, is so important!

Global Commerce

Technology allows businesses to sell products and services all over the world instantly.

- **Impact:** Online shopping (e-commerce) allows small businesses to reach customers globally. This increases competition and makes goods more available and sometimes less expensive for consumers.

Intentional vs. Unintentional Impacts

It's important to remember that not all impacts are planned!

Feature	Intentional Impact (Planned)	Unintentional Impact (Unexpected)
Definition	A change that was the goal of creating or using the technology.	A side effect or change that was not expected or planned by the creators.
Example: Fitness App	Helping a user track their health and lose weight (Personal)	Causing the user to constantly worry about their step count (Personal)
Example: Social Media	Connecting friends across the world (Social)	Creating echo chambers where people only see one side of an argument (Social)

When people design new technology, they usually focus on the positive **intentional impacts**. For example, a company designs a new energy-saving computer chip (intentional environmental impact).

However, new technology always has **unintentional impacts**. The same energy-saving chip still needs rare earth metals to be manufactured, contributing to e-waste (unintentional environmental impact).

Understanding both the good and the bad, the planned and the unplanned, helps us use technology responsibly! We must always ask: What are the full personal, social, environmental, and economic impacts of this new tech? 🤔

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 6 CS Chapter 5 Lesson Plans

View Grade 6 CS Chapter 5 Infographic

View Grade 6 CS Chapter 5 Video

View Grade 6 CS Chapter 5 Podcast

Copy Grade 6 CS Chapter 5 Slides

Copy Grade 6 CS Chapter 5 Self Guided Learning Worksheet

Copy Grade 6 CS Chapter 5 Scratch Activity Worksheet

Copy Grade 6 CS Chapter 5 micro:bit Activity Worksheet

Copy Grade 6 CS Chapter 5 Unplugged Activity Worksheet

Copy Grade 6 CS Chapter 5: The Impact of Technology: Personal, Social, Environmental, and Economic

Conclusion: You Are a Computer Scientist!

Wow! You've made it to the end of our Computer Science adventure. Give yourself a big high-five! You have learned so much about how computers work and how you can use coding to build and change the world around you.

Let's quickly review the super-cool concepts you've mastered.

The Power of Abstraction

Remember Abstraction? It's your secret weapon for solving tricky problems!

What is Abstraction?

Abstraction is all about simplifying the complex. Think of it as taking a really messy room and only showing someone the important parts, like the door and the window, while ignoring the mountain of laundry and toys (the unnecessary details!).

Abstraction Step	What you do
Determine details	Decide what's important and what's not.
Remove unnecessary details	Ignore the clutter!
Identify important information	Focus on the key stuff.
Generalize patterns	Find repeated ideas and simplify them.

Abstraction in Your Life

Abstractions are everywhere, making your daily life much easier. You don't need to know how the wires work inside a light switch, or the engine inside a car, to use them.

- Simple controls on appliances
- Light switches
- Steering wheels
- Apps on your phone

Designing with Code

You learned the basic building blocks of any program, called **structures**. These structures help you organize your instructions so the computer knows exactly what to do.

- **Sequence:** This is an ordered set of instructions. Step 1, then Step 2, then Step 3.
- **Conditionals:** These are "if-then-else" statements. They let the computer make decisions. *If* it is raining, *then* take an umbrella, *else* leave it at home.
- **Loops:** These tell the computer to repeat instructions over and over, saving you tons of time!

You even used a visual block-based language to put these structures into action and design your own computational artifacts!

Code's Impact on the World

Computer scientists, like you, design and code new artifacts to help with the world's needs and wants. Code powers everything from:

- Weather modelling
- Communication apps
- Automotive controls
- Medical research

Every piece of technology you use has code behind it, and that code has an **impact**.

Think about the different types of impacts:

Type of Impact	What it affects
Personal	How it changes one person's life.
Social	How it changes groups of people or communities.
Environmental	How it affects the natural world.
Economic	How it affects money and jobs.

Sometimes, impacts are intentional (you meant for them to happen), and sometimes they are unintentional (they happen by accident!). A big part of being a computer scientist is not just creating new things, but predicting the good and bad impacts they might have.

Your Future in Computer Science

You now have the knowledge and skills of a grade 6 computer scientist:

- You can **apply abstraction** to design.
- You can **identify examples of abstractions** every day.
- You can **discuss the role of design and coding in society**.
- You can **use a visual block-based language** to code.

- You can **discuss and predict the impacts** of technology.

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 6 CS Conclusion Self Guided Learning Worksheet

Copy Grade 6 CS Final Project Rubric

View Grade 6 CS Conclusion Infographic

View Grade 6 CS Conclusion Podcast

View Grade 6 CS Conclusion Video

Copy Grade 6 CS Conclusion Slides

Copy Grade 6 Computer Science Final Project

Copy Gr 6 CS Final Project Lesson Plan

Copy Grade 6 CS Conclusion: You Are a Computer Scientist!