TATA CONSULTANCY SERVICES | Discovery EDUCATION

IGNITE MY FUTURE

THINK SOLVE CREATE CONNECT

LESSON TITLE

Recycle and Reuse!

Guiding Question: How can we connect with each other?

SUBJECTS

Science
Engineering

COMPUTATIONAL THINKING PRACTICE

Developing and Using Abstractions

Creating Computational Artifacts

COMPUTATIONAL THINKING STRATEGY

Decompose

MATERIALS

Computers with Internet access and necessary requirements for the MIT Scratch web application

Real-Life Recycling and Reusing student capture sheet

Decomposing Code: Pros, Cons, and Solutions student capture sheet

Game Design student capture sheet

Ignite Curiosity

- How often do you use shortcuts or key commands when using a computer or smartphone?
- Why can't you recycle or reuse everything you use?
- When do you recycle and reuse in the real world?

Architects, engineers, and contractors use many tools to build new structures. While each new building is unique, many elements of all structures are the same. Instead of starting from scratch, architecture professionals use blueprints and plans from similar buildings to save time and avoid error. Like building a new house or corporate office, building new computer programs takes a lot of time and trial-and-error. In both situations, the computational thinking strategy of decomposition makes it possible to break apart the building process into manageable pieces and replicate those pieces over and over again to save time and prevent mistakes. In this lesson, students will use the computational thinking strategy of decomposing to build a new, unique computer game out of preexisting blocks of code. In **THINK**, students will act as software engineers developing a new game or phone app. Students will learn how using blocks can increase speed of programming, make programming more accessible, and reduce errors in coding. In **SOLVE**, students will use the Scratch visual programming tool to explore how games are created in blocks of code. Students will identify patterns of code blocks to generalize for use in their own game or app. Then, they will design their own game and decompose it into blocks of functionality (they may draw a storyboard, flowchart, or diagram, or they may write a list). In **CREATE**, students will use Scratch to create their own game or app. In **CONNECT**, students will discuss how coding professionals like software engineers and app developers share resources through open source code databases, code libraries, and widgets.

Students will be able to:

- **Examine** how to decompose complex problems into manageable sub-problems,
- **Analyze** patterns and common features between programs, and
- **Create** an original application based off of recycled blocks of code.

Students will act as software engineers developing code for a new game or phone app.

1 **Read** the following scenario to students:
Imagine you are a professional video game designer who has created a popular and successful game. Now, the company you work for wants you to create a follow-up that will be just as popular and successful—and you have to do it in half the time it took to create your first game! How can you meet this deadline and still produce a high-quality game? Let's see what you come up with!

Ask students if they think coders who create computer programs, video games, and phone apps write every piece of code from scratch. Ask them to imagine they are writing a message to a friend and can't copy and paste a quote, an image, or a link. They have to retype it all, character by character. That would be really annoying, right? Point out that computers have lots of functions that operate like copy and paste. They identify repetitive actions and simplify them to save us time. This is called **decomposing**. Visual programming editors allow users to drag and drop existing code blocks, create new code blocks from existing blocks, and map to a standard, text-based language that underlies the blocks.

2 **Lead** students to consider the importance of decomposing using the following guiding questions:

- What would happen if you had to create every piece of code in your game or app from scratch? (The coding would take too long, and you would miss your deadline.)
- How could decomposing help video game designers meet their deadlines? (copy simple and repetitive functions from existing games/a code block library)
- What elements do you see in video games that suggest the use of decomposing? (similar character templates, quests, health/energy/time bars, etc.)
- How would visual programming editors make it easier to decompose code? (Dragging and dropping is easier than copying and pasting.)

3 **Distribute** the Real-Life Recycling and Reusing student capture sheet to demonstrate that decomposing is not limited to the computer world. Ask students to think of other duplication tools they have encountered and add them to the list. Point out that there are many other uses for the tools listed on the handout. Have students select one of the tools and come up with at least one other example of a situation where it would be used.

4 **Distribute** the Decomposing Code: Pros, Cons, and Solutions student capture sheet. Point out that while the focus of this lesson has been on the advantages of decomposing tools, they also have disadvantages. Have the students write down at least five advantages and disadvantages of decomposing in coding, and encourage them to think about ways they could compensate for those shortcomings when working on their project.

5 **Challenge** students to identify and summarize the problem that needs to be solved. Remind them to consider what elements they can take from a code library or block-coding program and to consider the pros and cons of this approach.

Students will explore how games are created in blocks of code in the Scratch or Snap! visual programming tools.

Teacher Note: Now that students have explored decomposing tools in the physical and virtual worlds, ask them to consider how to use these tools to create a new video game or phone app. Tell students about goIT, an experiential, immersive coding and app development program for Canadian students.

Middle and High school students get hands-on technology education, working in teams to tackle icon-based programming languages and design challenges as they collaborate to build a robot. The program is run and funded by **TATA Consultancy Service (TCS), a leading global IT services, consulting, and business solutions organization. It most recently held its annual competition in Toronto.**

To find a coding program near you, visit Canada Codes Funded Initiatives.

1. **Introduce** students to the Scratch application. Explain that Scratch is a block-coding tool that makes coding easier, faster, and more accessible by providing a simple visual interface that enables coders to see blocks of code, choose the ones they want, and drag and drop them into a program. Ask students why a visual drag-and-drop interface makes coding easier and more accessible (coders do not have to learn a complex coding language). Demonstrate the program by showing examples. Teachers can access examples of Scratch via this link: https://scratch.mit.edu/starter_projects/.

2. **Encourage** students to discuss which blocks of code would be good candidates for decomposition. What blocks of code could be generalized? Remind them of the discussion about repeated elements in video games and to consider those elements as possible candidates. For example, a specific quest in a video game could be broken down into its basic story elements (for example, introduction of the protagonist, description of the quest's goal, description of the challenges the protagonist must overcome, etc.).

3. **Distribute** the Game Design student capture sheet. Have students brainstorm, either alone or in small groups, to come up with the game or app they want to build. Once they've chosen their project, have them outline its design on the worksheet, then deconstruct it into blocks of functionality

   Outline format options to include the following:

   - Storyboard
   - Flowchart
   - Diagram
   - List

Students will use the block-coding platform Scratch to create their own game or app.

Teacher Note: Be sure to allow enough time for students to explore the Scratch program, create their game or app, and try out and assess one another's products as a way of opening discussion around what worked, what could be improved, and how decomposing affected the building process.

1. **Introduce** students to Scratch, answering any questions they may have about the platform or about the project in general.

2. **Have students build their program,** following the design they created on the worksheet and using the block-coding platform. Then, have students test one another's programs and offer comments on what did and did not work, make suggestions for improvement, and identify where they see the use of block coding in the program.

3. **Summarize** by inviting students to note where they used block coding in their own program, review the comments and suggestions made by other students, and consider what blocks of code they could use from this program should they be asked to build another app or game. Tie this into the original scenario (a programmer has been asked to create a follow-up to a popular game on a much shorter timeframe), and ask students if their perspective on the scenario has changed, and why it has or has not.

Select one of the strategies listed below to help students answer these questions:

- **How do this problem and solution connect to me?**
- **How do this problem and solution connect to real-world careers?**
- **How do this problem and solution connect to our world?**

1. **Write** the three questions on PowerPoint or flip chart slides and invite students to share out responses.
2. **Display** pieces of chart paper around the room, each with one question written on it. Ask students to write down their ideas related to the questions on each sheet.
3. **Assign** one of the questions to three different student groups to brainstorm or research, and then share out responses.
4. **Invite** students to write down responses to each question on a sticky note, and collect them to create an affinity diagram of ideas.

How does this connect to students?

Students likely use more decomposing tools than they realize, including physical tools such as rubber stamps and copy machines and virtual tools such as copy/paste functions in word processing programs and cloning tools. Some students might even use games such as Super Mario Maker to create custom video game levels.

How does this connect to careers?

Computer Instructors use decomposing tools such as block code and code libraries to provide their students greater access to coding.

Computer Designers, including game designers and compiling designers who translate code from one language to another, use decomposing tools to code more rapidly and efficiently.

Robotics Engineers can use decomposing tools to repeat robotic code snippets within whole coded instructions to repeat identical actions.

How does this connect to our world?

Decomposing tools are used to increase the speed and efficiency of repetitive tasks in everything from typing emails and texts to customer service. For example, if you order the same thing every time you go to a restaurant, the staff can save time by pulling up the information from a saved order you made before and duplicating that information to create a new order. Or, if you write a lot of documents for your job, you can save time by creating a template for the letters and a database from which you can copy addresses.

One concern about decomposing tools is that they can remove the thought or effort behind repeated actions, leading to errors if the action is not identical. For example, if the restaurant duplicates your order and you want something different, the restaurant may not make the change. There is a reason "rubber stamping" refers to approving a request without reading its details. For example, if you copy the wrong address from a database, you could send an important letter to the wrong person. Additionally, AutoFill functions don't account for address or phone number changes.

Find more easy-to-implement resources to integrate computational thinking practices into your classroom by visiting **ignitemyfutureinschool.ca**

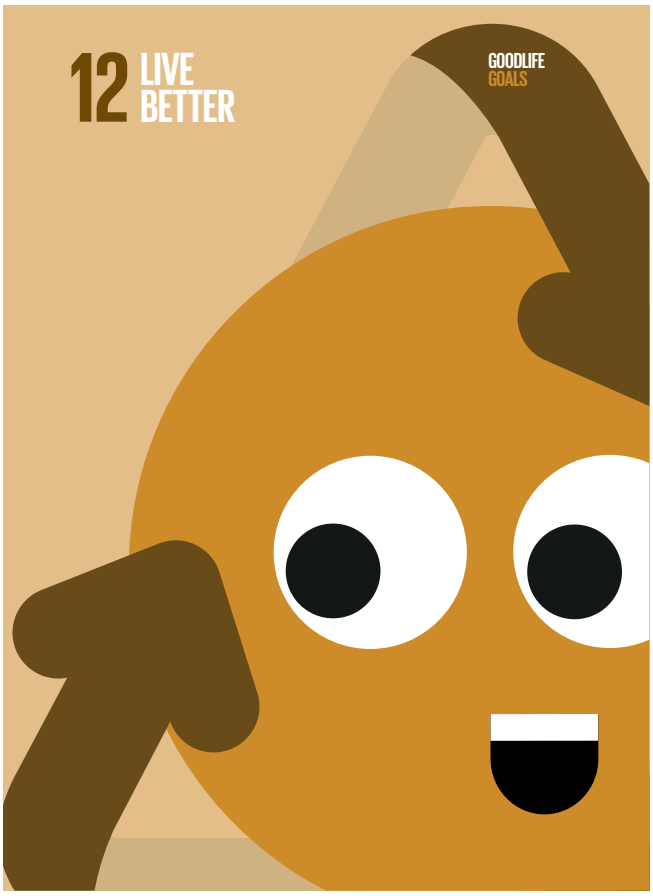
Curriculum Connections

UNITED NATIONS SUSTAINABLE DEVELOPMENT GOALS

12 RESPONSIBLE CONSUMPTION AND PRODUCTION

"For the goals to be reached, everyone needs to do their part: governments, the private sector, civil society and **people like you**."
–The United Nations

"The Sustainable Development Goals are the blueprint for a better future. And together we can reach them. By following the Good Life Goals, we can all help make tomorrow better than today. Let's do this! #GoodLifeGoals"



Source:
The Good Life Goals by Futerra Sustainability Communications Ltd and 10-Year Framework of Programmes on Sustainable Lifestyles and Education Programme is licenced under CC BY-ND 4.0.

Find more easy-to-implement resources to integrate computational thinking practices into your classroom by visiting **ignitemyfutureinschool.ca**

Global Competencies

CMEC (Council of Ministers of Education, Canada) Pan-Canadian Global Competencies Descriptions

Highlighted sections apply to this lesson

Global Competency	Definition	Student Descriptors
Collaboration	Collaboration involves the interplay of the cognitive (including thinking and reasoning), interpersonal, and intrapersonal competencies necessary to participate effectively and ethically in teams. Ever-increasing versatility and depth of skill are applied across diverse situations, roles, groups, and perspectives in order to co-construct knowledge, meaning, and content, and learn from, and with, others in physical and virtual environments.	Students participate in teams by establishing positive and respectful relationships, developing trust and acting co-operatively and with integrity. Students learn from and contribute to the learning of others by co-constructing knowledge, meaning, and content. Students assume various roles on the team, respect a diversity of perspectives, and address disagreements and manage conflict in a sensitive and constructive manner. Students network with a variety of communities/groups and use an array of technology appropriately to work with others.
Communication	Communication involves receiving and expressing meaning (e.g., reading and writing, viewing and creating, listening and speaking) in different contexts and with different audiences and purposes. Effective communication increasingly involves understanding both local and global perspectives, societal and cultural contexts, and adapting and changing using a variety of media appropriately, responsibly, safely, and with regard to one's digital footprint.	Students communicate effectively in different contexts in oral and written form in French and/or English through a variety of media. Students communicate using the appropriate digital tools and create a positive digital footprint. Students ask effective questions to acquire knowledge, listen to understand all points of view, voice their own opinions, and advocate for ideas. Students gain knowledge about a variety of languages and understand the cultural importance of language.

Global Competencies cont.

Highlighted sections apply to this lesson

Global Competency	Definition	Student Descriptors
Global Citizenship and Sustainability	Global citizenship and sustainability involve reflecting on diverse worldviews and perspectives and understanding and addressing ecological, social, and economic issues that are crucial to living in a contemporary, connected, interdependent, and sustainable world. It also includes the acquisition of knowledge, motivation, dispositions, and skills required for an ethos of engaged citizenship, with an appreciation for the diversity of people, perspectives, and the ability to envision and work toward a better and more sustainable future for all.	Students understand the ecological, economic, and social forces, their interconnectedness, and how they affect individuals, societies, and countries. Students take actions and responsible decisions that support quality of life for all, now and in the future. Students recognize discrimination and promote principles of equity, human rights, and democratic participation. Students understand Indigenous traditions and knowledge and its place in Canada, learn from and with diverse people, develop cross-cultural understanding, and understand the forces that affect individuals, societies, and nations. Students engage in local, national, and global initiatives to make a positive difference. Students contribute to society and to the culture of local, national, global, and virtual communities in a responsible, inclusive, accountable, sustainable, and ethical manner. Students as citizens participate in networks in a safe and socially responsible manner.

Global Competencies cont.

Highlighted sections apply to this lesson

Global Competency	Definition	Student Descriptors
Critical Thinking and Problem Solving	Critical thinking and problem solving involve addressing complex issues and problems by acquiring, processing, analysing, and interpreting information to make informed judgments and decisions. The capacity to engage in cognitive processes to understand and resolve problems includes the willingness to achieve one's potential as a constructive and reflective citizen. Learning is deepened when situated in meaningful, real-world, authentic experiences.	Students will solve meaningful, real-life, complex problems by taking concrete steps to address issues and design and manage projects. Students will engage in an inquiry process to solve problems as well as acquire, process, interpret, synthesize, and critically analyse information to make informed decisions (i.e., critical and digital literacy). Students will see patterns, make connections, and transfer what they have learned from one situation to another, including in real world applications. Students will construct, relate, and apply knowledge to all domains of life such as school, home, work, friends, and community. Students will analyze the functions and interconnections of social, economic, and ecological systems.
Innovation, Creativity and Entrepreneurship	Innovation, creativity, and entrepreneurship involve the ability to turn ideas into action to meet the needs of a community. The capacity to enhance concepts, ideas, or products to contribute new-to- the-world solutions to complex economic, social, and environmental problems involves leadership, taking risks, independent/unconventional thinking and experimenting with new strategies, techniques, or perspectives, through inquiry research. Entrepreneurial mindsets and skills involve a focus on building and scaling an idea sustainably.	Students formulate and express insightful questions and opinions to generate novel ideas. Students contribute solutions to complex economic, social, and environmental problems or to meet a need in a community in a number of ways including; enhancing concepts, ideas, or products through a creative process, taking risks in their thinking and creating, making discoveries through inquiry research, and by hypothesizing and experimenting with new strategies or techniques. Students demonstrate leadership, initiative, imagination, creativity, spontaneity, and ingenuity in a range of creative processes and motivate others with an ethical entrepreneurial spirit.

Global Competencies cont.

Highlighted sections apply to this lesson

Global Competency	Definition	Student Descriptors
Learning to learn and to be self-directed and self-aware	Learning to learn and to be self-directed and self-aware, means: becoming aware and demonstrating agency in one's process of learning, including the development of dispositions that support motivation, perseverance, resilience, and self-regulation. Belief in one's ability to learn (growth mindset), combined with strategies for planning, monitoring and reflecting on one's past, present, and future goals, potential actions and strategies, and results. Self-reflection and thinking about thinking (metacognition) promote lifelong learning, adaptive capacity, well-being, and transfer of learning in an ever-changing world.	Students learn the process of learning (metacognition) (e.g., independence, goal-setting, motivation) and believe in their ability to learn and grow (growth mindset). Students self-regulate in order to become lifelong learners and reflect on their thinking, experience, values, and critical feedback to enhance their learning. They also monitor the progress of their own learning. Students develop their identity in the Canadian context (e.g., origin and diversity) and consider their connection to the environment. They cultivate emotional intelligence to understand themselves and others. They take the past into account to understand the present and approach the future. Students develop personal, educational, and career goals and persevere to overcome challenges to reach these goals. They adapt to change and show resilience to adversity. Students manage various aspects of their lives: physical, emotional (relationships, self-awareness), spiritual, and mental well-being.

Find more easy-to-implement resources to integrate computational thinking practices into your classroom by visiting **ignitemyfutureinschool.ca**

Real-Life Recycling and Reusing

Tool	Medium	Purpose	Example of Use
Rubber stamp	Rubber and ink	Duplicating a simple image, word, or phrase	Marking files "Approved" or "Rejected"
Copy machine	Light-based machinery	Duplicating a printed image or page of text	Copying a page from a library book for note-taking purposes
Sculpture mold	Silicone or rubber	Duplicating a three-dimensional object	Producing a line of action figures
Copy/paste function	Text-based computer programs	Selecting and duplicating text in a text-based artifact	Inserting a hyperlink into a document
Cloning tool	Image editing programs such as Photoshop	Selecting part of an electronic image and copying it to another part of that image	Editing a digital photo to create a meme (such as face-swapping)
Visual coding interface	Block-based coding programs such as Scratch	Building a computer program from existing blocks of code	Designing a new app
Open source code	Computer program	Code that is available to anyone to use or modify without needing to ask permission	Improving the coding of an Android phone app

Decomposing Code: Pros, Cons, and Solutions

Pros	Cons	Possible Solution
Frees time from having to code simple and repetitive elements from scratch each time	End product could be boring and repetitive.	Use block coding only for the most basic elements, then elaborate so the end product stands out.
Helps prevent typos	Errors from the decomposed code could be copied over.	QC check decomposed code as carefully as original code.

Game Design Worksheet

1. **Use this space** to design your game or app.

2. **Note** opportunities to use block codes in your design.

3. **Comments:**