DeCoDe

How to Use **Design Thinking** & **Computational Thinking** to Solve Any Problem

Learn how to harness the creative power of divergent and convergent thinking to change the world, solve problems and develop innovative solutions.

For use in classrooms Grade 3 and above.

bsd education
www.bsd.education

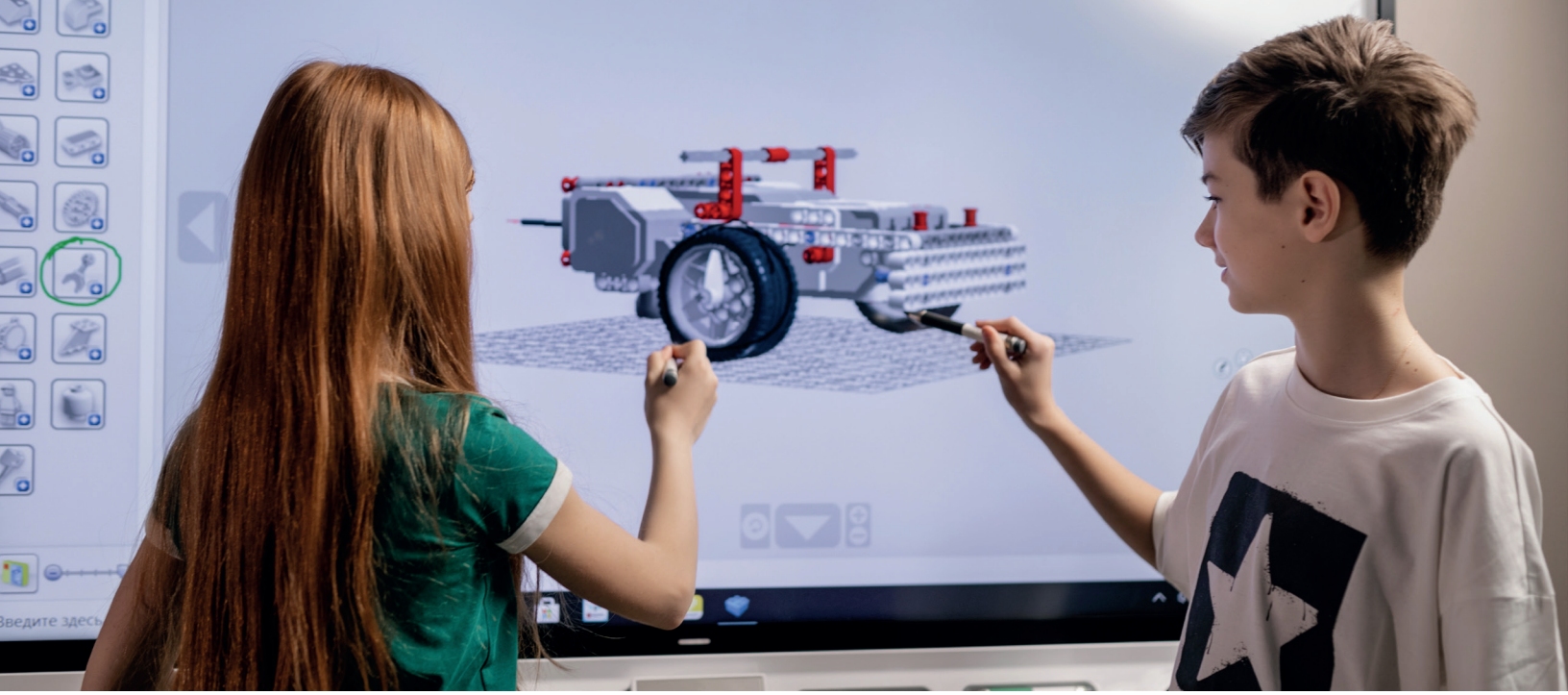
Design Thinking & Computational Thinking are both Creative Processes

What is Design Thinking?

Design Thinking is a process used in the world of design to create innovative solutions to problems and to design products. Design thinking is a powerful human-centered approach which requires the designer to empathize with the user's perspective and needs.

What is Computational Thinking?

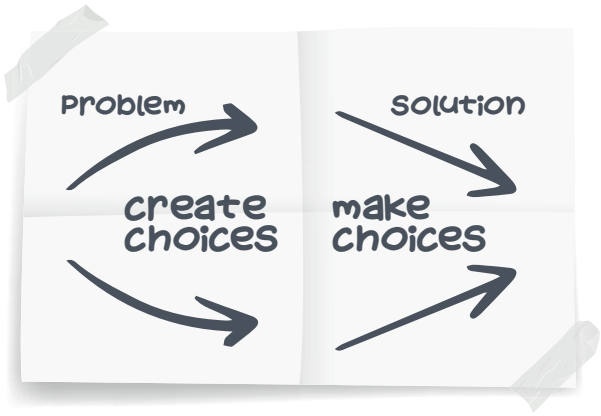
Computational Thinking is a way of taking possible solutions to a problem and carefully breaking the problem down into manageable chunks that can be easily repeated, similar to how a computer program uses logic to solve a problem.



<. DeCoDe />

The combined creative power of Design Thinking & Computational Thinking

Design Thinking is a type of **divergent thinking** that involves generating potential ideas and possible solutions using the power of empathy. Computational Thinking is a **convergent thinking** process that involves narrowing down the ideas into actionable designs that can be accomplished by breaking down the problem into smaller chunks.



Divergent Thinking refers to a creative process of generating original ideas and new possibilities.
Convergent Thinking is the process of finding concrete and useful solutions to problems.

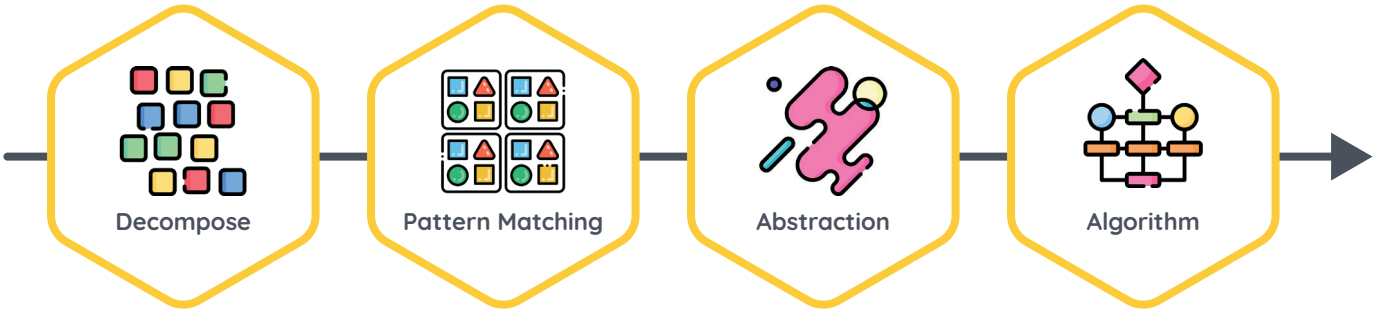
Design Thinking Process

Design Thinking has five phases:

- **Empathize** by engaging with users and learning about them as much as possible.
- **Define** the users' problems and needs.
- **Ideate** solutions by creating innovative ideas.
- **Prototype** a scaled-down, lightweight solution to the defined problem.
- **Test** solutions with actual users.

Computational Thinking Process

Computational Thinking has 4 phases that help break down problems into manageable steps that lead towards an actionable solutions.



- **Decomposition** - Break down a complex problem into smaller parts
- **Pattern Matching** – Look for similarities within a problem.
- **Abstraction** – Leave out irrelevant details to focus only on the important information.
- **Algorithm** - Devise a step-by-step solution (or rules to follow) to solve the problem.

<. DeCoDe />

How to combine the creative power of Design Thinking & Computational Thinking

Step 1

Start with Design Thinking.

Step 3

Complete the Computational Thinking Process to solve the problem. This solution will end up becoming the prototype of the Design Thinking process.

Step 2

Stop after the Ideate step of Design Thinking and enter into the Decompose step of the Computational Thinking process.

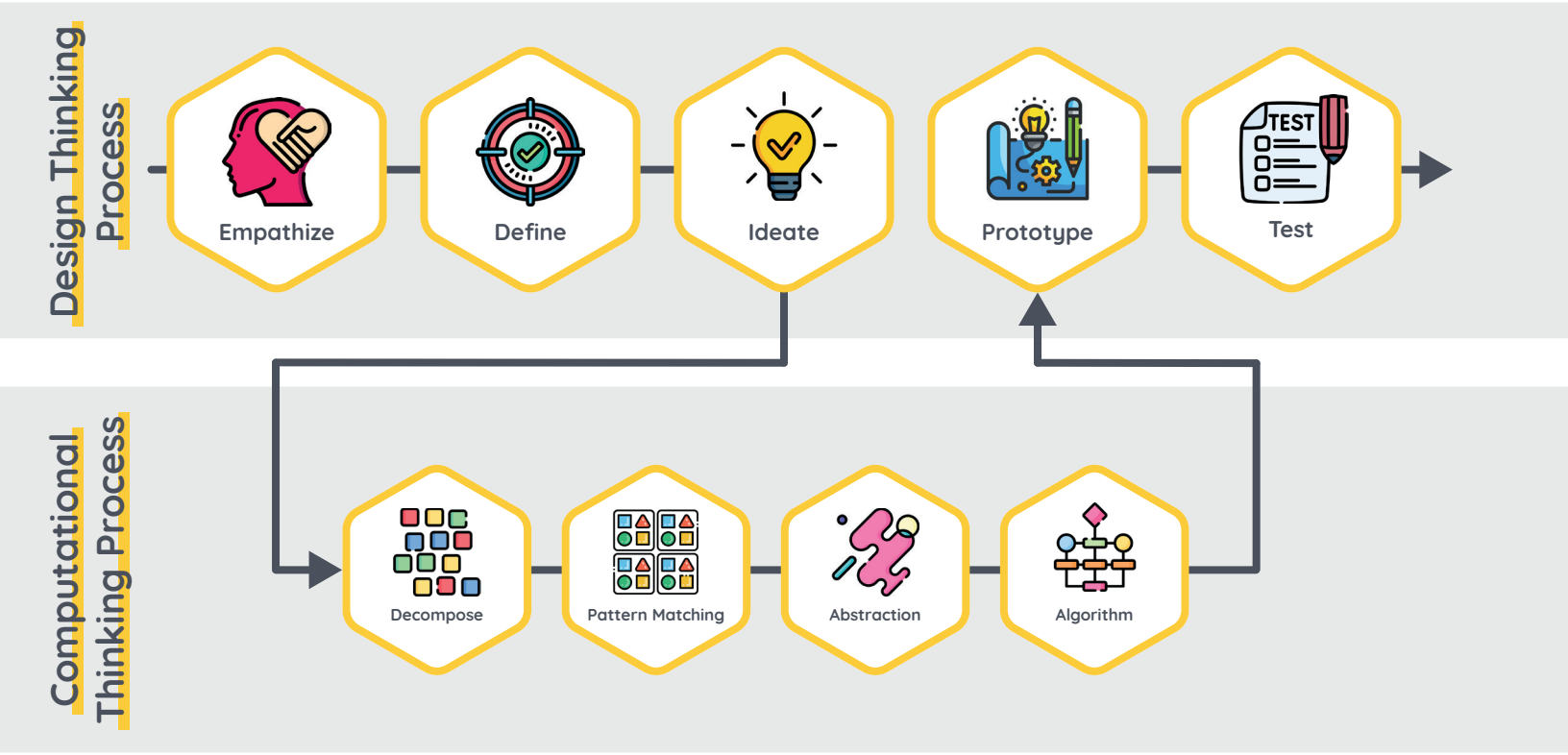
Step 5

Repeat the process if necessary, to continue refining the prototype or solution.

Step 4

Once the Computational Thinking Process is complete, move on to the rest of the Design Thinking process. (Prototype and Test steps)

Design Thinking Meets Computational Thinking



We call this process DeCoDe because the Computational Thinking process is sandwiched between both ends of the Design Thinking process. This powerful method of solving problems is also a way of decoding. To decode something means to break it down, to decipher it and make it useful in other ways. Any problem can be solved using this method.

Here is an example of how to use the DeCoDe method to solve a problem

Problem: My 13 year old sister is hungry and wants to eat lunch and it's my responsibility to cook lunch today.

DT Empathize: My sister really likes Indian food, pizza and sandwiches with a strong preference towards pizza. I am eating too, so I should cook something that we both like.

DT Define: I need to see what ingredients are available and of those, which do I know how to make.

DT Ideate: There are several options, but of the ingredients and my ability I will have to choose something that fits the time that I have, the ability that I have and what will taste good.

CT Decompose: Of the options, it seems like a sandwich is the best choice, but I will make it a grilled cheese sandwich because it is as close to a pizza that I know how to make.

CT Pattern Matching: Since I need to make 2 sandwiches, I can prepare the ingredients in a way that minimizes my preparation time.

CT Abstraction: The most important element is cooking the grilled cheese to perfection, and not burning the bread, so I will focus on that task because it yields the greatest result.

CT Algorithm: Cut cheese to size, butter the bread, cook slowly, serve on a plate with fresh salad like tomatoes and cucumber.

DT Prototype: Follow the steps of the algorithm to actually prepare the lunch.

DT Test: Eat and enjoy. Ask everyone what could be better about the meal.

<. DeCoDe /> Activity

Now that you know more about how to use the two creative methods, here is an example that you can try for yourself.

Problem: Alice needs to sign up for after school courses on Monday, Wednesday and Friday because her parents are working late. She would like to do 1 physical activities class, 1 art class and 1 computer class. Alice also has trouble remembering her schedule and needs daily reminders.

Here is the school schedule:

Monday	Tuesday	Wednesday	Thursday	Friday
Football Painting	Martial Arts Poetry	Computer Coding Ceramics	Robotics Drama	Swimming Dance

Let's use the power of Design Thinking and Computational Thinking to help Alice solve the problem

DT Empathize: In this step we would ask Alice what her favorite activities were. Let's say that Alice likes all of the classes the school is offering, but dancing is her least favorite option. What other questions would you ask Alice?

DT Define: Let's design an after school schedule for Alice based on her likes, interests and availability. Let's also think of ways to help remind Alice about her schedule. Do we need to define any other elements?

DT Ideate: Come up with 2 possible schedule plans for Alice and 2 possible ways to remind her.

CT Decompose: Let's pick one of the options that you created. Write down which option you picked.

CT Pattern Matching: Are there any patterns that could help us design a useful solution?

CT Abstraction: What information isn't important that we don't need to worry about?

CT Algorithm: What is your schedule for Alice and how will she remember?

DT Prototype: Describe your prototype plan and reminder for Alice.

DT Test: Share your plan with someone else to get feedback. Write down the person's feedback.

<. DeCoDe /> Worksheet

Now that you know how the process works, you can use it to solve any problem. Here is a worksheet for you to try:

Problem: What problem are you trying to solve?

DT Empathize: Ask questions to the person (or group) you are solving the problem for to better understand their needs and requirements.

DT Define: Are there any special requirements or limitations that you should consider, such as how much time you have, how much budget you have or any other restrictions.

DT Ideate: Make a list of possible solutions, these could be drawings, sketches, words, videos or any other medium.

CT Decompose: Choose one of the options and commit to working on it by considering the factors from the DT Define step.

CT Pattern Matching: Are there any elements of the solution that have patterns that can be helpful in solving the problem? Can those patterns help you be more efficient?

CT Abstraction: What is the most important element of the solution? Focus on that!

CT Algorithm: Write out the steps needed to solve the problem. You can draw pictures, make a flowchart or describe the steps however you want.

DT Prototype: Follow the steps of the algorithm to actually create your solution. This is where creativity comes into action!

DT Test: Share your solution with the person (or people) that you designed for and reflect on the whole process together.

Next Steps

Now that you have learned how to harness the creative power of divergent and convergent thinking, you are ready to start your first coding project. Design Thinking and Computational Thinking are the skills that programmers and designers use to create websites, mobile apps, video games, virtual reality worlds, artificial intelligence, and more.

If you are a teacher and want to dive deeper into using the DeCoDe method with guided coding projects, then **BSD START** is where you should begin.

https://bsd.education/bsd-start/

Learn more about incorporating coding into your classroom with BSD Education at **www.bsd.education**

Let's Build Something Different Together

Start integrating future-ready skills today

For more information contact us at

info@bsd.education