Overview
Juniors learn more about algorithms, abstraction, functions, and variables through activities where they create mad glibs and craft suncatchers.

First, girls learn that abstraction is one of the most important skills for a computer scientist to understand. It simplifies problems and prevents unnecessary repetition. A good coder uses abstraction just about every time she creates a program. Girls will analyze stories for differences and abstract them to create fun mad-libs.

Then, Juniors explore computational thinking by creating suncatchers using string and beads. They will learn how this process is like creating an algorithm with patterns, variables, and functions (pieces of code that you want to use over and over again).

Notes for Volunteers:

Use The Talking Points (But Make Them Your Own): In each session, you'll find suggested talking points under the heading “SAY.” Some volunteers, especially new ones, find it helpful to follow the script. Others use the talking points as a guide and deliver the information in their own words. Either way is just fine.

Be Prepared (It’s What Girl Scouts Do!): Each meeting includes a “Prepare Ahead” section that includes a materials list and what kind of set-up is required. Read it in advance so you have enough time to gather supplies and enlist help, if needed.

Use Girl Scouts’ Three Processes: Girl-led, learning by doing, cooperative learning — these three processes are the key to making sure Juniors have fun in Girl Scouts and keep coming back.

“Learning by doing” and “cooperative learning” are built into this Journey, thanks to the hands-on activities and tips. You’ll also find specific “keep it girl-led” tips in the meeting plans. They’ll help you create an experience where Juniors know they can make choices and have their voices heard.

Solve Big Problems Step By Step: On this Journey, Juniors will do hands-on activities to learn how computer programmers think through problems. They’ll learn to follow and create algorithms, break big problems down into smaller ones, and persist when faced with challenges.

You can help Juniors think this way! Encourage them to keep trying when their first few approaches to solving a problem don’t work. Tell them that they can solve any problem...
Think Like a Programmer pt. 2

if they break it down in smaller ones. And remind them that they can use those skills in their daily lives as well.

**Leave Time For The Closing Ceremony:** If Juniors are having fun doing an activity, you may be tempted to skip the Closing Ceremony so they can keep going — but the Closing Ceremony is absolutely key to their learning. Here’s why:

When Juniors leave a meeting, they’ll remember how much fun it was to plant a seed, make a suncatcher or play a game of "Programmer Says." However, they may not realize that they just learned how algorithms work — unless you tell them. When you do that, you turn a *hands-on* activity into a *minds-on* activity. During the Closing Ceremony, you can connect the dots for girls by:

- Pointing out how they acted as programmers. (For example: They used an algorithm to plant a seed or they created an algorithm to teach a skill to others. They struggled a bit with a challenging activity — but they persisted. Now they know that they can solve hard problems if they keep trying. They worked together to solve problems.)
- Reminding Juniors that they are *already* programmers — and that it’s fun to solve problems using programming.
- Letting them know that they have what it takes to continue exploring STEM.

These simple messages can boost girls’ confidence and interest in STEM — and end the meeting on an upbeat note!

**Tell Your Troop Story:** As a Girl Scout leader, you’re designing experiences that girls will remember their whole lives. Try to capture those memories with photos or videos. Girls love remembering all they did — and it’s a great way for parents to see how Girl Scouting helps their girls.

And please share your photos and videos with GSUSA by emailing them to STEM@girlscouts.org (with photo releases if at all possible!).

**Program Pairing:** The Junior Digital Photographer and Entertainment Technology badges go well with this Journey!

**Prepare Ahead (Roughly 100 minutes)**

1. **Watch five videos (26 minutes)**

   **Mad Glibs overview video** (0:29): code.org/girlscouts/MadGlibs/OverviewVideo

---

© 2017 GSUSA. All rights reserved. Not for commercial use. This Material is proprietary to GSUSA and may be used, reproduced and distributed exclusively by GSUSA staff, councils, Girl Scout volunteers, service units and/or troops solely in connection with Girl Scouting.

We are pleased to provide these materials solely to our staff, councils, Girl Scout volunteers, service units and/or troops to use in connection with Girl Scouting. These materials are protected under copyright law and we appreciate your compliance in not reproducing and/or redistributing these materials without our written permission. If you have any doubts as to whether your proposed use violates our copyright law rights, please email us at: permissions@girlscouts.org with your question.
Think Like a Programmer pt. 2

This video gives a brief overview of Mad Glibs for Activity 3: Solving Challenges with Computational Thinking.

**Unplugged Activity: Mad Glibs video** (0:44):
[link](code.org/girlscouts/MadGlibs/ActivityVideo)

The video introduces girls to Mad Glibs in Activity 3: Solving Challenges with Computational Thinking. This video may be shown to the girls before starting the activity. This is optional. You may not have the wi-fi connection or the time to show the video.

**CS Fundamentals Unplugged: Computational Thinking** demo video (9:10):
[link](code.org/girlscouts/ComputationalThinking/DemoVideo)

This video provides a demo of Mad Glibs for Activity 3: Solving Challenges with Computational Thinking and tips for facilitating the activity.

(Note to Volunteers: This video was filmed in a classroom setting. Of course, Girl Scout troop meetings are not like school. Your girls won’t sit at desks, and you may need to shorten your set-up.)

Listen for the main points you want to make with girls:

- **Computational Thinking** takes big, complicated problems and breaks them up into smaller, more manageable pieces.
- Abstraction, decomposition, pattern matching, and algorithms are all tools of Computational Thinking that can be used to solve problems.
- Abstraction is pulling out specific differences to make one solution work for multiple problems.
- In this activity, patterns and abstraction helped us to create an algorithm or solution that will work for multiple problems.

**CS Fundamentals Unplugged: Functional Suncatchers** demo video (8:11)
[link](code.org/girlscouts/FunctionalSuncatchers/DemoVideo)

This video provides a demo of Functional Suncatchers for Activity 3: Solving Challenges with Computational Thinking and tips for facilitating the activity.

(Note to Volunteers: This video was filmed in a classroom setting. Of course, Girl Scout troop meetings are not like school. Your girls won’t sit at desks, and you may need to shorten your set-up.)

© 2017 GSUSA. All rights reserved. Not for commercial use. This Material is proprietary to GSUSA and may be used, reproduced and distributed exclusively by GSUSA staff, councils, Girl Scout volunteers, service units and/or troops solely in connection with Girl Scouting.

We are pleased to provide these materials solely to our staff, councils, Girl Scout volunteers, service units and/or troops to use in connection with Girl Scouting. These materials are protected under copyright law and we appreciate your compliance in not reproducing and/or redistributing these materials without our written permission. If you have any doubts as to whether your proposed use violates our copyright law rights, please email us at: permissions@girlscouts.org with your question.
Listen for the main points you want to make with girls:

- How is a program used to replicate a task?
- How do functions help us to simplify programs?
- How do you use variables within functions and programs? What is their purpose?

**Course 3- Functional Suncatchers overview video** (6:38)  
code.org/girlscouts/FunctionalSuncatchers/OverviewVideo

This video gives you an overview of Functional Suncatchers for Activity 3: Solving Challenges with Computational Thinking and tips for facilitating the activity. It highlights how the activity relates to patterns, functions, and variables in programs.

**Unplugged – Functional Suncatchers activity video** (1:01)  
https://www.youtube.com/watch?v=d1MdyeXy0v0

The video introduces girls to Functional Suncatchers for Activity 3: Solving Challenges with Computational Thinking and how the concepts of functions and variables that can be used in programming. This video may be shown to the girls prior to starting the activity. This is optional. You may not have the wi-fi connection or the time to show the video.

2. Review vocabulary (2 minutes)

This meeting includes the following vocabulary:

- **Algorithm** — a list of steps that allow you to complete a task
- **Program** — instructions (or an algorithm) that can be understood and followed by a machine
- **Function** — a piece of code that you can easily call over and over again
- **Variable** — a placeholder for a piece of information that can change
- **Decompose** — to break a hard problem up into smaller, easier ones
- **Pattern** — a theme that is repeated many times
- **Abstraction** — removing details from a solution so that it can work for many problems
- **Computational Thinking** — the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.
See the Junior Think Like a Programmer Journey Glossary for more vocabulary and examples.

3. Read through this guide and its Meeting Aids (15 minutes)

This will help you get familiar with the flow of the meeting.

Read the following handouts (found in the Meeting Aids section):

Junior Think Like a Programmer Journey Materials List: Each meeting has its own materials list, but you can use this handout if you like to do all your supply shopping at one time. It includes all materials needed for the entire Journey.

Junior Think Like a Programmer Journey Glossary: This is a list of words that Juniors may not know and how to define them.

Think, Pair, Share: These facilitation tips will help you to make sure that every girl’s voice is heard during brainstorming activities.

Take Action Guide: This handout explains the difference between Take Action and Community Service. It also includes tips to make a project sustainable and Take Action project ideas that you and your troop can use as inspiration.

4. Gather materials (60 minutes)

Gather materials using the Materials List for this meeting. If your meeting location doesn’t have a flag, bring a small one that Juniors can take turns holding or hang in the room.

Prior to girls arriving for the meeting, organize the snacks by type into different bowls/cups for Activity 1: As Girls Arrive: Snack Algorithms.

Get Help from Your Family and Friends Network

Your Friends and Family Network can include:

- Girls’ parents, aunts, uncles, older siblings, etc.
- Other volunteers who have offered to help with the meeting
Ask your Network to help:
- Make snacks
- Bring art supplies (beads, spacers, etc.) for Functional Suncatchers.

**Award Connection**
Juniors earn two awards:
- Think Like a Programmer award
- Take Action award

Juniors will earn both awards following the completion of the Take Action project and Journey in **Think Like a Programmer PT. 6**.

*(Note to Volunteers: You can buy these awards from your council shop or on the Girl Scouts’ website.)*

**Meeting Length**
90 minutes
- The times given for each activity will be different depending on how many Juniors are in your troop.
- There is no snack time scheduled in these meetings, but there are 15 minutes of “wiggle room” built in for snacks or activities that run long.
- Give Juniors 10- and 5-minute warnings before they need to wrap up the last activity so you’ll have time for the Closing Ceremony.

Juniors learn more about algorithms, abstraction, functions, and variables through activities where they create mad libs and craft suncatchers.

First, girls learn that abstraction is one of the most important skills for a computer scientist to understand. It simplifies problems and prevents unnecessary repetition. A good coder uses abstraction just about every time she creates a program. Girls will analyze stories for differences and abstract them to create fun mad-libs.

Then, Juniors explore computational thinking by creating suncatchers using string and beads. They will learn how this process is like creating an algorithm with patterns, variables, and functions (pieces of code that you want to use over and over again).
Think Like a Programmer pt. 2

Materials List

Activity 1: As Girls Arrive: Snack Algorithms

(Note to Volunteers: For this activity, girls will create algorithms for snacks. We’ve provided several options, so please choose one or another alternative that works for your girls’ dietary restrictions.)

- Bowls or cups
- Option 1: Various small snacks like crackers, pretzels, marshmallows, chocolate chips, dried fruit, etc. that could go into trail mix.
- Option 2: Crackers, spreads, and toppings to create cracker bite snacks.
- Option 3: Celery, spreads, and topping to create Ants on a Log.
- Paper
- Pencils

Activity 2: Opening Ceremony: Programmers to the Rescue!

- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

Activity 3: Solving Challenges with Computational Thinking

- Mad Glibs Abstraction Worksheet (one for each girl)
- Markers, pens, or pencils
- Pens, Pencils, & Scissors
- Functional Suncatchers Skills Sheet (one for each group)
- Create a suncatcher to use as an example for the activity
- Prepare a program and two skills with blank steps on a large paper or blackboard for the girls to help you fill in. (Sample can be found on the Functional Suncatchers Skills Sheet)
- Optional: Computer, tablet or other device with ability to show girls the Unplugged: Mad Glibs activity and Unplugged – Functional Suncatchers videos
- Optional: Worksheet: Mad Glibs Assessment and Worksheet: Functional Suncatchers (one for each girl)
  An “assessment worksheet” sounds a lot like school, but girls will probably see this as a fun puzzle page. If there’s time, girls could do the activities in the meeting or you could give each girl a copy to take home. Perhaps they’d like to
show their families what they learned about programs, debugging, functions, and variables by doing the worksheet together.

For each functional suncatcher:
- One foot of string, thread, or fishing line
- 2-4 beads
- 2-4 other accessories (buttons, hoops, spacers)
- One special bead, prism, button, or girl-made sun charm

Activity 4: Closing Ceremony: Brainstorming Our Take Action Project
- List of Juniors’ Take Action ideas from Think Like a Programmer PT. 1
- Take Action Guide handout

Awards
Juniors do not receive any awards in this meeting.

Detailed Activity Plan

Activity 1: As Girls Arrive: Snack Algorithms

Time Allotment
10 Minutes

Materials

(Note to Volunteers: For this activity, girls will create algorithms for snacks. We’ve provided several options, so please choose one alternative that works for your girls’ dietary restrictions.)

- Bowls or cups
- Option 1: Various small snacks like crackers, pretzels, marshmallows, chocolate chips, dried fruit, etc. that could go into trail mix.
- Option 2: Crackers, spreads, and toppings to create cracker bite snacks.
- Option 3: Celery, spreads, and topping to create Ants on a Log.
- Paper
- Pencils
Steps
Prior to girls arriving, organize the snacks by type into different bowls/cups.

Welcome Juniors, and have them create an algorithm for their ultimate snack bite.

**SAY:**
*What’s your favorite type of snack? What ingredients taste good together?*

*The ingredients for your snack are in these bowls. Choose the ingredients you like to create your own personal snack.*

*Then write an algorithm so others can make it, too.*

*For example, I like to take a cracker (that’s step one), put a piece of cheese on it (that’s step 2), then top it with a slice of cucumber (that’s step three). If you use my algorithm, I guarantee you’ll have a tasty snack!* 

**Activity 2: Opening Ceremony: Programmers to the Rescue!**

**Time Allotment**
15 Minutes

**Materials**
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

**Steps**
Recite the Pledge of Allegiance and the Promise and Law.

Conduct any troop business.

Talk about how programmers help others by solving complex problems.

**SAY:**
*Has anyone ever had a problem that seemed so big they couldn’t solve it alone? How did you tackle the problem? Did it help to break it into smaller?*

**Girls may say:** I had a big project for school, I had a performance, etc.
Last time, you were able to break up the steps to help your friend recreate an image in tangrams without knowing what they were making.

Explain decomposition to Juniors.

**SAY:**
You broke the problem of not knowing how to recreate the tangram down into something much simpler by creating a list of steps to make the image. Often, big problems are just lots of little problems stuck together.

You “decomposed” the activity into smaller tasks, letting each step seem a little easier on its own. This is called decomposition.

Can you think of any other time in life you might have “decomposed” a problem?

**Girls might say:** I had a school project I made a plan for, I made a recipe, etc.

**You decompose big problems into smaller ones every day!**

Tools of Computational Thinking, like decomposition and algorithms, can help you to figure out how to solve problems that no one has already taught you how to solve!

Programmers also use their imaginations to solve problems. They define a need by seeing what problems there are in the world that they can create programs to improve. They brainstorm ideas to design and build the program in teams. Then they test the program and share it with others to test, debug it, and improve it.

Review the following definitions, asking girls for their ideas for each:

- **Algorithm** — a list of steps that allow you to complete a task. For example, a recipe is an algorithm.
- **Program** — instructions (or an algorithm) that can be understood and followed by a machine
- **Debugging** — finding and fixing problems in your algorithm or program

**SAY:**
Today, we are going to learn about abstraction as we create fun mad-glibs and then create an algorithm to make a suncatcher.
Activity 3: Solving Challenges with Computational Thinking

Time Allotment
40 Minutes

Materials
- Mad Glib Abstraction Worksheet (one for each girl)
- Markers, pens, or pencils
- Pens, Pencils, & Scissors
- Functional Suncatchers Skills Sheet (one for each group)
- Create a suncatcher to use as an example for the activity
- Prepare a program and two skills with blank steps on a large paper or blackboard for the girls to help you fill in. (Sample can be found on the Functional Suncatchers Skills Sheet)
- Optional: Computer, tablet or other device with ability to show girls the Unplugged: Mad Glib activity and Unplugged – Functional Suncatchers videos
- Optional: Worksheet: Mad Glib Assessment and Worksheet: Functional Suncatchers (one for each girl)

An “assessment worksheet” sounds a lot like school, but girls will probably see this as a fun puzzle page. If there’s time, girls could do the activities in the meeting or you could give each girl a copy to take home. Perhaps they’d like to show their families what they learned about programs, debugging, functions, and variables by doing the worksheet together.

For each functional suncatcher:
- One foot of string, thread, or fishing line
- 2-4 beads
- 2-4 other accessories (buttons, hoops, spacers)
- One special bead, prism, button, or girl-made sun charm

Steps

Set Up. (5 minutes)

Explain abstraction to Juniors.

SAY:
So, what did you have for waffles this morning? (Your girls may look perplexed.)
No one? Okay, what did you have for toast yesterday? (You may start to get some hands raising, and people eager to share and agree because they want to relate, but possibly not because they understand.)

See what I was doing there? I was too specific about what I had yesterday, and that made it harder for everyone else to understand me. What could I have said that more people would have understood?

Let girls answer. At some point, they’ll start to come up with the idea of using “breakfast” in place of the actual food that was consumed.

SAY:
The word ‘breakfast’ is like a variable that we use to hold a space for whatever it is we ate this morning.

By taking the specific word out and replacing it with ‘breakfast,’ we are using abstraction to make something work for multiple people.

Abstraction is a way of pulling out specific differences to make one solution work for multiple problems.

Ask the troop to give you some examples of other ways that they might use abstraction to create different sentences that can be understood by more people.

Let girls answer. The idea of lunch and dinner will most-likely come up. Challenge them to brainstorm if there is anything not food related?

Optional: Show the Unplugged Activity: Mad Glibs video here.

Create Mad Glibs. (10 minutes)

Hand out the Mad Glibs Abstraction Worksheet, one for each girl, and explain the activity.

SAY:
This started as a specific story about one thing, but we used abstraction to turn some of the specific words into blanks, and now the story can be about lots of things.

What could you make your story about?

Let girls answer.
Each girl should spend around 10 minutes to fill in the blanks in Story 1 of their worksheets with words to create something fun to share.

Once each girl has filled in their story, form a Friendship Circle.

One at a time, each girl can share her story.

**SAY:**

*Your stories are all unique, but they started from the same thing!*

*Using computational thinking, you recognized a pattern, you were able to “abstract out”, or ignore the details that make things different, and use the general framework to find a solution that works for more than one problem.*

*Abstraction is removing details from a solution so that it can work for many problems.***

If there is extra time, girls can recreate the story once more using the space under Story 2 on their worksheet and share in pairs or as a larger group.

**Introduce Functional Suncatchers. (5 minutes)**

Show your suncatcher to the girls, and explain how you made it.

**SAY:**

*What do you all think about my suncatcher?*

**Girls may say:** Your suncatcher is pretty, I like the beads, I like the charms.

*You will be making a very similar, but slightly different suncatcher today!*  
*If I wanted to tell you how I made my suncatcher, how could I do that?*

**Girls may say:** Write down the steps, tell me how, take a picture, etc.

*I could create an algorithm to tell you exactly how I made my suncatcher.*

Explain variables to Juniors and that they will be using them to make their own suncatchers.

**SAY:**

*The materials may be different for each of your suncatchers, but the steps to make them will be pretty much the same.*
[Pointing to a bead] For example, my suncatcher has beads like this. Yours may be a different color or even a different shape, but it is still called a "bead".

[Pointing to a spacer] This is what my space looks like. Yours may be a different size, different texture, or maybe a different material altogether, but it will still be called a "spacer".

[Pointing to a special charm] Finally, we will all have a "special charm". Yours might be a large bead, a handmade ornament, or even a random item from the room, but we all will call it a "special charm."

The differences between the types of beads, spacers, or charms are examples of variables. Can anyone tell me what a variable is?

**Girls may say:** Things that change, etc.

A variable is a placeholder for a piece of information that can change.

**Optional:** Show the Unplugged – Functional Suncatchers video [here](#).

**Learn about Functions. (10 minutes)**
Show Juniors the sample suncatcher, and explain how they will create their suncatchers.

**SAY:**
Let me tell you how I made this suncatcher.

(You may use the following SAY:

**Example SAY:** First, I put a bead on the string, then I tied a knot. I put another bead on a string, and tied another knot. Then, I put a spacer on the string and tied another knot. After that, I did it all again. I put a bead on the string, then I tied a knot. I put another bead on a string, and tied another knot. Then, I put a spacer on the string and tied another knot. Finally, I put on the special charm, and tied one last knot.

That's a lot of steps, right? I'll give you a sing-songy way to remember the order.

**optional:** It helps to have a bit of a rhythm with the words as you go through.

**Example:**

...
“Bead, knot, bead, knot, spacer, knot.
Bead, knot, bead, knot, spacer, knot.
Special charm, final knot.”

Give each girl a **Functional Suncatchers Skills Sheet** and markers, pens, or pencils.

**SAY:**
*To help, I will also write down the steps to create this program, so that everyone has the directions in front of them to make their own suncatchers.*

Use the paper or chart you prepared on the board with the Program and two Skill set (as seen in the **Functional Suncatchers Skills Sheet**), and begin to write the steps in the "Program" area as the girls shout out instructions. Write one instruction per line.

As the girls give instructions and you write them on the paper/board, girls should also copy the instructions on to their **Functional Suncatchers Skills Sheet**.

**Example:**

**PROGRAM**

1. Bead
2. Knot
3. Bead
4. Knot
5. Spacer
6. Knot

By now, the girls should be noticing an issue (there are more instructions than lines). If they don’t, **ASK:**

- *How many instructions do we have left to go through?*
- *How many lines do we have left?*
- *What should we do?*

**Girls may say:** There aren’t enough lines to make the suncatcher, we could add more lines, etc.

Point out that there are two “extra” sections on the Skills Sheet.

**SAY:**
*There are two “extra” sections on the Skills Sheet. You can use these to combine steps so that you can write one name to call them all in the Program section.*

*For example, everything that you put into the top section, can be used all at once by using “Skill 1” in the Program instructions; this is a function.*
A function is a piece of code that you can easily use over and over again to help simplify programs.

Sometimes, when a problem has lots of little pieces, you will notice that the pieces have something in common. If they don’t, then they may at least have some similarities to some pieces of another problem that has been solved before. If you can spot these patterns, understanding your pieces gets much easier.

What sort of patterns did you see?

Girls might say: Bead knot bead, etc.

A pattern is a theme that is repeated many times.

You can find these patterns and create a function that can be called to repeat the pattern over and over! Then you had enough space on your list for all the steps.

Challenge the girls to fill out their Functional Suncatchers Skills Sheet in a way that makes sense and allows them to fit the entire sequence under the “Program” list in some way.

Give Juniors about five minutes to complete their sheet (more for younger girls).

While you circulate the room, if you see that a girl’s program is incorrect, help her to “debug” and correct it.

Build Functional Suncatchers. (10 minutes)

Hand out supplies to the girls. Give each girl:
- One foot of string, thread, or fishing line,
- 2-4 beads,
- 2-4 other accessories (buttons, hoops, spacers), and
- One special bead, prism, button, or girl-made sun charm.

Point out that different people may have different supplies, and they may all be different than the ones you used for the sample suncatcher.

SAY:
If we have different supplies, should that stop us from sharing the program (instructions) to create the suncatcher? Why or why not? Does anyone remember what we call these differences?
Girls may say: No because even if you use different beads, you will still make a suncatcher, the suncatcher will still work, it's okay for everyone’s suncatcher to look different, etc.

Right! Every suncatcher will be a little different, which is part of the fun.

Do you remember what we call these kinds of differences? (Variables.)

A variable is a placeholder for a piece of information that can change. We can treat the words (Bead, Spacer, Knot, etc.) as placeholders or variables for whatever items we are using that fit those descriptions.

Let the girls make their suncatchers, using the program and functions they made.

Permission for use of Code.org activities is provided by Code.org, a non-profit dedicated to giving every student in every school the opportunity to learn computer science. See www.code.org.

Activity 4: Closing Ceremony: Brainstorming Our Take Action Project

Time Allotment
10 Minutes

Materials
- List of Juniors’ Take Action ideas from Think Like a Programmer PT. 1
- Take Action Guide handout

Steps
Have Juniors form a Friendship Circle and share their suncatchers.

Remind Junior how they used computational thinking to solve problems in the activities today.

SAY:
You did a great job using computational thinking to create fun stories and programs to make your suncatchers today.
Think Like a Programmer pt. 2

How did you do it? Were there any special skills you used to make your suncatcher? What about in the Mad Glibs activity?

Girls may say: We abstracted the Mad Glibs, we made an algorithm, we used strong string, etc.

It sounds like you used the steps of computational thinking, just like programmers!

In both activities, you were posed with a problem (how to fill in the Mad Glib and how to make a suncatcher).

You decomposed, or broke down, these problems into smaller steps to find solutions.

You did this by looking at patterns, abstracting out details, and creating functions from the patterns you saw.

You brought all of this together and created algorithms which solved the original challenges.

Have Juniors brainstorm Take Action projects, and remind them how Take Action can use computational thinking.

SAY:

Just like programmers, you can use computational thinking to plan your Take Action project. Programmers work in teams, just like you, to test their algorithms over and over again to find solutions that work for everyone.

Last meeting you talked about how Take Action projects and programming help make the world a better place. Did anyone come up with some new Take Action project ideas?

Write down Juniors’ ideas.

(Note to Volunteers: Check out the Take Action Guide handout in the Meeting Aids if Juniors need help with ideas. Bring the list of ideas Juniors come up with to the next meeting.)

End the meeting with a Friendship Squeeze.
Sometimes you want to do certain instructions over and over again. That’s where functions come in handy! Group all repeated instructions into one place, give them a simple name, then you can call that entire group at the same time just by using the name you gave it.

What if you want to do something over and over, but don’t know what supplies you’ll be working with ahead of time? This is the perfect place for variables! Variables are just placeholder words that you can put into your program so that you know where your *actual* supplies are supposed to go, once you know what they are.

**Directions:**

1) Take a program that contains several sets of identical instructions.

2) Move one or more of the sets of identical instructions into the “Skills” areas of the Skills Sheet.

3) Rewrite the original program, using the skill names instead of actually writing out the group of instructions that the skills describe.

**New Words!**

**Function**

*Say it with me: Func-shun*

*A piece of code that you can easily call over and over again*

**Variable**

*Say it with me: Vayr-ee-ah-buhl*

*A placeholder for a piece of information that can change*
**Functional Instructions**
Skills Sheet

**SKILL 1**
1) ________________
2) ________________
3) ________________
4) ________________
5) ________________
6) ________________

**SKILL 2**
1) ________________
2) ________________
3) ________________
4) ________________
5) ________________
6) ________________

**PROGRAM**
1) ______________________
2) ______________________
3) ______________________
4) ______________________
5) ______________________
6) ______________________

Revision 140702.1a
The Girl Scout Promise
On my honor, I will try:
To serve God and my country,
To help people at all times,
And to live by the Girl Scout Law.

The Girl Scout Law
I will do my best to be
honest and fair,
friendly and helpful,
considerate and caring,
courageous and strong, and
responsible for what I say and do,
and to
respect myself and others,
respect authority,
use resources wisely,
make the world a better place, and
be a sister to every Girl Scout.
Think Like a Programmer Journey

Take Action Guide

What’s the difference between a community service project and a Take Action project?

Community Service makes the world better by addressing a problem “right now.” For example, collecting cans of food for a food pantry feeds people “right now.” Gathering toys for a homeless family shelter makes kids happy “right now.” Providing clothing and toiletries to people after a fire or flood helps them “right now.” These acts of kindness are important ways to help people — right now.

Take Action encourages girls to develop a project that is sustainable. That means that the problem continues to be addressed, even after the project is over. Sustainability simply means coming up with a solution that lasts.

For example, girls might want to do something about trash in a local park. If they go to the park and pick up trash, they’ve solved the problem for today — but there will be more trash to pick up tomorrow.

Instead, girls could explore why there’s so much trash. Here’s what they might discover:

1. There aren’t enough trash cans in the park.
2. The trash cans are hard to find.
3. People have to walk out of their way to throw away trash because of where the cans are placed.
4. People don’t realize the importance of putting trash in the trash cans.

Here’s how girls might address these issues:

• Issues 1 – 3: Make a presentation to the city council to report on their findings and suggest adding more trash cans or moving them to more visible or convenient positions.
• Issue 4: Create a public awareness campaign that encourages people to use the trash cans instead of littering.
• Variation: Older girls may want to design interactive garbage cans that make tossing your trash fun. Do an online search for “the fun theory” or “the world’s deepest bin” to see this in action.
What are the steps of a Take Action project?

Girls team up to:

- Identify a problem
- Come up with a sustainable solution
- Develop a team plan
- Put the plan into action
- Reflect on what they learned

Keep It Girl-Led: Girls should actively participate in each step in order for this to be girl-led. Younger girls will need more guidance, but they can and should decide as a team what problem they want to address.

How do girls make their project sustainable?

Here are three ways to create sustainable change:

1. Make your solution permanent.
2. Educate and inspire others to be part of the change.
3. Change a rule, regulation or law.

How can I help girls come up with Take Action Ideas?

Next are some specific examples you can use to help girls understand what sustainable Take Action projects look like.

Keep It Girl-Led: These examples are intended to give a sense of what a Take Action project could look like. Please do not choose a project from this list for girls to do! Instead, guide them to brainstorm ideas, get feedback, and come up with a plan. Girls will learn key leadership skills, such as decision-making, compromise, conflict resolution, and teamwork, when their Take Action project is girl-led.
Computer Science/STEM Take Action Ideas

Issue: Some girls think computer science is hard or boring or just for boys.
• **Solution:** **Educate and inspire others.** Create a girls’ coding club that meets at lunchtime or recess. Teach other girls how to play with tangrams or learn algorithms by making functional suncatchers.

Issue: Some kids think computer science is too hard to understand.
• **Solution:** **Educate and inspire others.** Make a video to explain algorithms, using fun examples like baking a cake, planting a flower or giving directions. Show it to your class at school or to a group of friends.

Issue: More kids need to know that how computer programmers can help others and make the world a better place
• **Solution:** **Educate and inspire others.** Do some research about people who used code to help others, and then create a video or slideshow to show at your school.

Issue: Not everyone knows about women who changed the world using their knowledge of computer science.
• **Solution:** **Educate and inspire others.** Research the “hidden figures” in your community. They might be women who have helped shape history, like those portrayed in the movie Hidden Figures. Or you might want to profile computer science teachers who have made a difference by mentoring and encouraging girls. You could create a display about their accomplishments for a library or community center or make a video about them and show it at school.

Issue: More people need to know how exciting and fun STEM can be.
• **Solution 1:** **Educate and inspire others.** Create a list of great books, movies and documentaries that focus on STEM. Make copies for teachers to hand out or make posters for the school library.
• **Solution 2:** **Educate and inspire others.** Create a short play based on one of the books and perform it for your class or school.
Other Ideas for Take Action

**Issue:** More kids need to know that engineering is a fun, creative way to help others.

- **Solution 1:** Educate and inspire others. For show-and-tell, explain what you’ve learned about how engineers help others, then lead a design challenge activity with your class.

- **Solution 2:** Make it permanent. Partner with a teacher or principal to create an “engineering space” at school where kids can make prototypes and share ideas for new inventions. Put out a call for donations of recyclable materials or cheap prototyping supplies (cardboard boxes, tape, string, paper towel tubes, etc.) to stock the space.

**Issue:** It’s hard for new students to meet people and make friends at school.

- **Solution:** Make it permanent. Design and build “buddy benches.” Partner with the school to have the benches installed on the playground so kids who want to make new friends can find each other.

**Issue:** Parents often run their engines outside the school as they wait to pick up or drop off their children, which pollutes the air.

- **Solution:** Change a rule, regulation or law. Make a presentation to the school board or administrators about why this is a problem and suggest a new rule that makes the pick-up/drop-off area a “no idling” zone.

**Issue:** We could conserve water if more people collected rain water and used it to water plants.

- **Solution 1:** Make it permanent. Make rain collection devices for family or friends that can be installed in their yards. Give them a list of different ways to use rain water and how they’re helping the Earth.

- **Solution 2:** Educate and inspire others. Create a handout, video tutorial, or show-and-tell presentation about how to make a rain collection device, how to use rain water and how that helps the Earth.

**Issue:** The local park doesn’t have a swing for children with disabilities.

- **Solution:** Make it permanent. Make a presentation to the city council explaining the problem and offering to use troop money from the cookie sale to help pay for the swing.

- **Extra Inspiration:** Do an online search for “How One Brownie Troop Became Social Entrepreneurs.”)
**Issue:** There’s no sidewalk along a street near the elementary school, which makes it dangerous for children to walk home.

- **Solution:** Make it permanent. Make a presentation to the city council about the problem and suggest that they build a sidewalk. (Note: Even if the council doesn't vote to create a sidewalk, the girls have earned their Take Action award because they came up with a sustainable solution and took action through their presentation.)

- **Extra Inspiration:** Do an online search for “Girl Scout Brownies Convince City Hall to Build Sidewalk.”

**Issue:** There have been several accidents at a busy intersection that doesn't have a stoplight.

- **Solution:** Make it permanent. Research the number of accidents and make a presentation to the city council, asking that they have a stoplight installed.

**Issue:** The local shelter is having a hard time getting rescue animals adopted.

- **Solution:** Educate and inspire others. Use your photography skills to create pet portraits for the shelter’s web site. Use your writing skills to craft heart-warming bios for each portrait.

**Need more ideas?**

Check out [Girls Changing the World](https://www.girlscouts.org) on the GSUSA web site. Girls post their Take Action and Bronze/Silver/Gold Award projects on this site. You can search by project topic or grade level. (And after the troop has done their project, please post it so they can inspire other girls!)
33 Ways to Take Action!

Make your solution permanent.
1. Make and install something outside (benches, bird houses, dog run, ropes course, sensory trail for children with disabilities, Little Library, etc.)
2. Plant something (butterfly garden, tree, wind chime garden, etc.)
3. Make something inside (Maker Space, reading room, etc.)
4. Create a collection (children’s books, children’s hospital or family shelter, oral histories for town museum, etc.)
5. Advocate for building a permanent community improvement (sidewalk, bridge, park, streetlights, stoplight, etc.)

Educate and inspire others to be part of the change.
6. Do a show-and-tell
7. Create a poster campaign
8. Perform a skit
9. Make a “how to” handout
10. Draw a comic
11. Give a speech
12. Write and perform a song
13. Make an animated movie
14. Make a live-action movie
15. Make a presentation
16. Create a workshop (perhaps in partnership with a local business or organization) to teach a skill such as coding, camping, canoeing, robotics, sewing, car care, healthy eating, gardening, home repair, budgeting, etc.
17. Create a workshop to teach others about healthy living (exercise, nutrition, mental health, etc.)
18. Create a social media campaign
19. Make video tutorials to teach a skill
20. Organize an email campaign
21. Organize a petition
22. Organize an event (concert, play, poetry slam, art exhibit, sporting event, field day) to raise awareness about an issue
23. Make a “playbook” to help others follow your lead (how to mentor robotics teams, organize a workshop or event, advocate to city council, create an online petition, change a law, etc.)
24. Make an app that helps people take action on an issue
25. Create a web site
26. Write an op-ed or letter to the editor of a newspaper or magazine
27. Start a blog

Change a rule, regulation or law.
28. Make a presentation to your school principal
29. Make a presentation to your school board
30. Make a presentation to your city council
31. Speak up at your representative’s town hall meeting
32. Create an online petition
33. Advocate for a law with your state government
Think Like a Programmer Journey

Glossary for Juniors

Juniors may not know some of the words used on this Journey. Here are definitions you can share with them:

**Computational thinking** is the thought process involved in solving a problem and expressing its solution(s) in a way that a computer—human or machine—can effectively carry it out.

An **algorithm** is a list of steps that you can follow to finish a task. A recipe is an example of an algorithm; it tells you how to cook a dish by following step-by-step instructions.

A **program** is an algorithm that has been coded into something that can be run by a machine.

**Debugging** is finding and fixing problems in your algorithm or program.

A **function** is a piece of code that you can easily call over and over again.

A **variable** is a placeholder for a piece of information that can change.

**Decomposition** is when you break a hard problem up into smaller, easier ones.

A **pattern** is a theme that is repeated many times.

**Abstraction** is removing the details from a solution so that it can work for many problems.

**Innovation** is a new or improved idea, device, product, etc.

A **prototype** is a sketch of an idea or model for something new. It’s the original drawing from which something real might be built or created.

In computer science, a **conditional** is a statement that tells a computer how to act in specific situations., i.e. IF this happens, THEN the computer does this.
Think Like a Programmer Journey

Materials List

Think Like a Programmer 1

**Activity 1: As Girls Arrive: Create Your Own Code**
- Paper
- Pens or pencils

**Activity 2: Opening Ceremony: Jump Into Computational Thinking!**
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

**Activity 3: Tangram Algorithms**
- **Tangram Set & Algorithm Card Images Pack** (one for each girl)
- Scratch paper for writing algorithms or building images
- Markers, pens, or pencils
- Scissors
- Optional: Sets of tangrams to use as example
- Optional: Computer, tablet or other device with ability to show girls the [Unplugged – Tangram Algorithms](link) video
- Optional: If your troop has never used Tangram pieces, you can choose to do an example for them or even have an entire Tangram activity. Search the internet for activities that girls can play in reality (using real tangrams) or play online.
- Optional: **Worksheet: Tangram Algorithms** (one for each girl)
  An “assessment worksheet” sounds a lot like school, but girls will probably see this as a fun puzzle page. If there’s time, girls could do the activities in the meeting or you could give each girl a copy to take home. Perhaps they’d like to show their families what they learned about programs, debugging, functions, and variables by doing the worksheet together.

**Activity 4: Closing Ceremony: Programmers, Awards & Take Action**
- Take Action Guide

Think Like a Programmer 2

**Activity 1: As Girls Arrive: Snack Algorithms**
(For this activity, girls will create algorithms for snacks. We’ve provided several options, so please choose one or another alternative that works for your girls’ dietary restrictions.)
- Bowls or cups
- Option 1: Various small snacks like crackers, pretzels, marshmallows, chocolate chips, dried fruit, etc. that could go into trail mix.
- Option 2: Crackers, spreads, and toppings to create cracker bite snacks.
- Option 3: Celery, spreads, and topping to create Ants on a Log.
- Paper
- Pencils

**Activity 2: Opening Ceremony: Programmers to the Rescue!**
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law
Think Like a Programmer Journey

Materials List

Think Like a Programmer 2 (continued)

**Activity 3: Solving Challenges with Computational Thinking**

- Mad Glibs Abstraction Worksheet (one for each girl)
- Markers, pens, or pencils
- Pens, Pencils, & Scissors
- Functional Suncatchers Skills Sheet (one for each group)
- Create a suncatcher to use as an example for the activity
- Prepare a program and two skills with blank steps on a large paper or blackboard for the girls to help you fill in. (Sample can be found on the Functional Suncatchers Skills Sheet)
- Optional: Computer, tablet or other device with ability to show girls the Unplugged: Mad Glibs activity and Unplugged – Functional Suncatchers videos
- Optional: Worksheet: Mad Glibs Assessment (one for each girl)
  An “assessment worksheet” sounds a lot like school, but girls will probably see this as a fun puzzle page. If there’s time, girls could do the activities in the meeting or you could give each girl a copy to take home. Perhaps they’d like to show their families what they learned about programs, debugging, functions, and variables by doing the worksheet together.
- Optional: Worksheet: Functional Suncatchers (one for each girl)
  An “assessment worksheet” sounds a lot like school, but girls will probably see this as a fun puzzle page. If there’s time, girls could do the activities in the meeting or you could give each girl a copy to take home. Perhaps they’d like to show their families what they learned about programs, debugging, functions, and variables by doing the worksheet together.

For each functional suncatcher:

- One foot of string, thread, or fishing line
- 2-4 beads
- 2-4 other accessories (buttons, hoops, spacers)
- One special bead, prism, button, or girl-made sun charm

**Activity 4: Closing Ceremony: Brainstorming Our Take Action Project**

- List of Juniors’ Take Action ideas from Think Like a Programmer 1
- Take Action Guide

Think Like a Programmer 3

**Activity 1: As Girls Arrive: Tech Collages**

- Magazines and catalogs, tech or regular
- Scissors
- Glue sticks
- Construction paper
- Optional: Stickers, other things to add into the collages

**Activity 2: Opening Ceremony: Reviewing Our Take Action Ideas**

- Flag
- List of Take Action ideas from last meeting
Think Like a Programmer Journey

Materials List

**Think Like a Programmer 3 (continued)**

**Activity 2: Opening Ceremony: Reviewing Our Take Action Ideas (continued)**
- Index Cards (Slips of paper, post-its, or a whiteboard and marker)
- Pens
- Tape
- Optional: Poster Board with the Girl Scout Promise and Law

**Activity 3: Personal Innovations**
- Post-its or slips of paper and tape (at least one for each girl)
- Markers, pens, or pencils
- Poster paper for sharing innovations (one for each girl)
- **Personal Innovations Activity Guide** (one for each girl)
- Markers, pens, or pencils
- Tape to hang posters

**Activity 4: Closing Ceremony: Time to Decide on Take Action!**
- Juniors’ Take Action ideas on index cards.
- Optional: Computer/tablet or other device with ability to show girls the [Computer Science is Changing Everything](#) video

**Think Like a Programmer 4**

**Activity 1: As Girls Arrive: Innovate Your Take Action!**
- Paper
- Pencils, crayons and markers

**Activity 2: Opening Ceremony: Programming for a Better World**
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

**Activity 3: Designing Our Take Action Project**
- Large pieces of paper or poster boards
- Markers
- Post-It notes
- Pens/pencils

**Think Like a Programmer 5**

**Activity 1: As Girls Arrive: Improv with Conditionals**
- Notecards with different statements written on them from the **Conditional Examples Sheet** (at least two for each pair of girls)
Think Like a Programmer Journey

Materials List

Think Like a Programmer 5 (continued)

Activity 2: Opening Ceremony: If We Take Action, Then We Make a Difference!
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

Activity 3: Creating Our Take Action Project
- Any materials Juniors need for their Take Action project

Think Like a Programmer 6

Activity 1: As Girls Arrive: Get Ready to Celebrate!
- Girl Scout Promise and Law poster(s)
- Any items Juniors want to display (such as photos or videos from their Take Action project)
- Photos and videos from the Journey meetings
- Music system
- Decorations
- Snacks

Activity 2: Opening Ceremony: Welcome!
- Flag
- Optional: Poster Board with the Girl Scout Promise and Law

Activity 3: Awards Ceremony and Celebration
- Think Like a Programmer award
- Take Action award

(Note to Volunteers: You can buy these awards from your council shop or on the Girl Scouts’ website.)

Activity 4: Girl Survey
- If girls are taking the survey online: Laptop/tablet
- If girls are filling out the survey on paper: Copies of Girl Survey (pdf available in Meeting Aids) and pen or pencil
Brainstorming Tips: Think, Pair, Share

How to Run a Think, Pair, Share Activity:

Tell girls that they’re going to brainstorm answers to your question using “Think, Pair, Share.”

Lead girls through the basic steps by telling them they will:

1. **Break into small groups.**

2. **Listen to the question or prompt.**

3. **Think about their answers.**
   - Girls may want to write their answers down.
   - Twenty seconds should be enough time, since girls will need to sit quietly.

4. **Pair with other girls.**
   - Girls talk with one to three other girls (depending on group size), making sure everyone has a chance to share their answers. If there’s time, it’s OK for girls to ask questions about each other’s answers.
   - For pairs, 20 seconds should be enough time. If your troop enjoys discussion, consider extending this to 1 to 2 minutes.

5. **Share with the group.**
   - Girls share their answers with the larger group.
   - This can be completed in 20 – 30 seconds, but will run longer based on group size and how the group sharing is done.

There are two ways to set up group sharing:

- **Strongly Recommended:** One girl shares the best/most interesting/summary answer for the group. This approach is great if you’re running short on time. It also helps develop conflict resolution and compromise skills.

- **Optional:** Each girl shares her partner’s answer. This helps girls develop active listening skills, but will run longer because all girls are sharing.
Write a story using the Mad Glibs template below. Fill in the blanks with words to create something fun to share. Then, create a second story by writing another version on the lines at the bottom of the page.

---

**Story 1**

First you take your ____________ then add a layer of ____________ before you pour on a hearty dose of ____________ .

Next, press some ____________ down into the ____________ before covering with a sprinkle of ____________ .

That’s how we make a ____________ !

---

**Story 2**

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________

________________________
Below, you will find three sets of skills, and a program that calls them.

Use the New Program and the skills that go with it to figure out what the steps of the Original Program were. Fill out the steps of the Original Program appropriately.

**ORIGINAL PROGRAM**

1) ____________
2) ____________
3) ____________
4) ____________
5) ____________
6) ____________
7) ____________
8) ____________
9) ____________
10) ____________
11) ____________
12) ____________
13) ____________
14) ____________

**NEW PROGRAM**

1) ____________
2) ____________
3) ____________
4) ____________
5) ____________

**SKILL 1**

1) banana
2) face
3) smells
4) ____________
5) ____________

**SKILL 2**

1) cat
2) ____________
3) ____________
4) ____________
5) ____________

**SKILL 3**

1) one
2) stinky
3) ____________
4) ____________
5) ____________

**NEW PROGRAM**

1) Skill 3
2) Skill 2
3) Skill 3
4) Skill 1
5) Skill 2
The Mad Glib template that we used to make these stories has vanished! Look at the stories and figure out which words are supposed to be blanks, then recreate the template at the bottom of the page.

---

**Story 1**

Early last year, my mom gave me an old skateboard. She told me about the days when she would ride it from her school in her hometown. I tried to ride it once, but tripped over my shoelaces. It didn’t take long before I decided that it was best to leave the skateboarding to my mom.

**Story 2**

Sometime last year, my mom told me an old story. She told me about the days when she would hear it from her father in her childhood. I tried to tell it once, but tripped over my words. It didn’t take long before I decided that it was best to leave the storytelling to my mom.

---

Create new template here: