There are many readily available lessons, activities, and curricula on the internet, in books, and in the minds of educators. The challenge in creating lessons for your program is to sift through what is available and tailor it to meet your goals and the needs of your participants. Remember that in the true spirit of computer science, anything is hackable! Look to the ideas you find as inspiration and use the following tips and resources to tailor them to align with the core values of your program and to make them as engaging as possible for participants.

Click on a title below to jump to that section.

// SETTING GOALS
// CHOOSING ACTIVITIES
// ADAPTING FOR INCLUSIVITY
// CONNECTING TO SOCIAL ISSUES
// CREATING AN ENGAGING LESSON
SETTING GOALS

An important first step before scouring the internet for ideas for lessons is to take time to think about what goals you have for the program and the participants. The following questions will help you think through the various components. This information, combined with some logistical information around the number and age of participants as well as how many hours of instructional time you have, will set a solid foundation for choosing lessons that best meet your needs and those of your participants.

PROGRAM GOALS

For each time your program meets, think through what you want participants to learn in that time. What do you want them to be able to do, understand, feel, and know after your time together? What tools/platforms will enable them to do those things? And what themes or social issues will they engage with to help solidify their learning? (See Connecting to Social Issues for more on this.) You might have one or several goals; however, be sure to be realistic, taking into consideration the amount of time you have and the age of your participants. In addition, ensure they all connect to your program learning goal.

WHAT PARTICIPANTS WILL DO + HOW THEY WILL DO IT (APPLICATION/PLATFORM) + WHY THEY WILL DO IT (THEME/SOCIAL ISSUE) = PROGRAM GOAL

Examples:

- Participants will explore ways robots are used and connect issues in education to robot uses.
- Participants will increase their confidence in developing apps on App Inventor to help people locate fresh food in their community.
- Participants will be able to develop a simple game in Scratch based on issues around climate change.
PARTICIPANT LEARNING GOALS

In addition to the specific technology-related skills you hope participants learn, it is important to think about what other skills they will obtain and how you will measure this. We encourage you to think about the ways participants will gain:

- A growth in self-confidence in problem-solving, computing skills, and innovating/creating using technology
- An increased sense of belonging, and being more likely to believe they can succeed in learning and working in tech
- An increase in desire to engage in future IT/computing opportunities

Refer to the “Impact and Reflection” section for more ideas on how to collect this data from your participants.

Then, think about what you/your team will do to help participants reach these goals. Hands-on activities participants can relate to and opportunities to explore, create, and make mistakes will help participants learn best. Avoid activities where they are required to sit still and listen, especially for long periods of time.

Examples:

- Tinker/maker lab: Set up a variety of stations with tools/applications for participants to explore freely. You can set up time parameters and challenges for participants to complete.
- Group/pair work
- Participants as teachers: Pair participants who have more experience with those who have less, and/or have groups learning different skills and then teaching each other.
- Take it away from the screen: Although a great deal of computer science learning can be done in front of a screen, brainstorm ways that participants can learn concepts actively before they execute.
- Simulations/role play: Rather than tell participants how something works or how it connects, let them experience it. Set up simulations or role plays so participants can discover.
- Field trips: Give participants an opportunity to experience what IT/coding looks like rather than just watching a video or listening to people talk about it.
CHOOSING ACTIVITIES

The lessons and activities you choose will guide your entire program, and the platform/tool(s) you choose will be what your participants use during the lessons and activities. Some platforms will have an already-designed curriculum you can use, others will require you to design things yourself, and most will require some of both. The questions below act as a checklist of things to think about as you make your selections. Remember that if a particular lesson doesn’t have these components already, you can go back and modify parts so that it does. As a reminder, be sure to check out the “Activity Ideas” section on the Engage Students page if you need help finding activities for your program.

IS IT DEVELOPMENTALLY APPROPRIATE FOR THE AUDIENCE YOU ARE SERVING?

Although a certain program, robot, or social issue may be exciting to you, it might not always resonate with the age group in your program. Similarly, activities that might work well for participants in second grade might be too simple for those in eighth grade. It is important to consider some common characteristics of the age group in your program, and choose or design lessons and an application that will be most engaging for them. Think about lessons designed with multiple learning styles and abilities in mind, providing a variety of ways for participants to engage in learning.

ARE THE NECESSARY MATERIALS AND TECHNOLOGY/SOFTWARE ACCESSIBLE AND RELEVANT?

Do you know what background knowledge (if any) participants have with the materials and technology you plan to introduce? Consider where and how you will get the technology/software you need if you don’t already have access to it. If running the program virtually, think about how your participants will have access to this technology as well. Do the tools and materials used meet a range of accessibility needs (e.g., left-handed scissors, space for assistive devices or personal assistants, closed captioning)?
ARE THE LEARNING GOALS ACHIEVABLE IN THE TIME YOU HAVE?
If not, can you adapt the lessons and goals to be achievable? Some activities may have participants building complex apps to support their community over a 2-week time period, meeting 10 different times for 8 hours each day (a total of 80 hours). If your program is meeting 4 times for 4 hours each (a total of 16 hours), it is likely not realistic to expect participants will achieve the same goals. Consider adapting the goals to your schedule or choosing lessons that more closely aligns with your program characteristics. You will also want to make sure activities are appropriately sequenced and progressive, meaning that new skills or concepts are not introduced without prerequisite skills or concepts being covered first.

ARE YOU PREPARED TO FACILITATE THESE ACTIVITIES?
If not, what support do you need? Consider how comfortable you are with the activities and technology being used — would you need to spend time learning and practicing, or have you facilitated similar material before? Build in time to get comfortable with the activities and discuss any additional support you may need or want with your implementation team and partners. Remember to also consider implementing a peer mentor to assist you in facilitating. They may be more familiar with the content and can also better relate to the audience you are serving.

ARE THE LESSONS CULTURALLY RESPONSIVE?
Instead of only presenting information to your students, try building off of your participants’ life experiences and consistently bringing them into your lessons. Current, real world examples allow for deeper engagement and help make connections between individual, community, and global identities. It is also important to understand your own cultural identity, and its consequences. Be yourself with your participants and form honest, caring, and human relationships while holding everyone to equally high expectations. Just as you are not defined by one aspect of your identity, neither are your participants. View the intersectionality of various social categories such as race, class, gender, language, ability, etc. as assets. For additional tips on how you can engage a diverse range of learners (specifically girls) in computing by using a culturally responsive approach, check out NCWIT’s “How Can You Engage A Diverse Range of Girls in Technology?” resource.

DO THE LESSONS PRIORITIZE HANDS-ON LEARNING WITH OPPORTUNITIES TO EXPLORE AND CREATE?
If lessons are lecture-based or require participants to do a lot of sitting and listening, consider how you will make them engaging. Think about encouraging experimentation. Let participants know that there can be multiple solutions to any problem and that there is no one “right” way of problem solving.
ADAPTING FOR INCLUSIVITY

Since one of the core values of AspireIT is inclusivity, it is important to think about how you will make sure that everyone feels included and welcomed when planning your program. From the beginning, work towards creating a community where participants feel they belong. It is important for participants to learn while they’re in your program, and they will be able to do this more effectively if they feel like this is a place in which they are valued and celebrated for who they are. Dedicating time each day for your participants to get to know each other through icebreakers and other activities can increase their comfort level and sense of belonging.

You also want to be thoughtful and aware of the visuals and examples you gather as you design your lesson. These include: anything you put on walls, what you use in activities and stories, and the images and examples used in the platform/applications you’ve chosen. Ensure that your participants’ identities are represented and that you are presenting a diversity of experiences. Be sure the language used encourages a “growth mindset” and a sense of belonging. (For more information on “growth mindset,” see the Facilitation section.)

And remember that no matter how perfect your lesson plan may seem, having back-up plans is essential. It is rare that you will know the exact background and experience level that each of your participants brings to the table, which means you may have planned a lesson that is too easy or too much of a challenge. There will also be times when the technology breaks or participants are really engaged in one aspect and you don’t want to interrupt that flow. So, it’s important to remain flexible and have back-up activities (such as having LEGOs and other offline activities for participants to engage with as needed). These activities can also serve as a nice place for participants to engage if they are struggling, frustrated, or just need a “brain break.” Some days you might not stick to your agenda exactly, and that is OK. It is most important that participants are learning and having fun exploring IT/computing, not that you did everything you said you were going to do each day.

“The best advice I can offer to another (facilitator) is to have a backup plan. You never know how long an activity will take, when the campers will lose focus, or what activities the campers will find most challenging. It’s okay to help campers along and always have an activity ready as a plan B.

- PROGRAM FACILITATOR
CONNECTING TO SOCIAL ISSUES

Participants will be most engaged if what they are learning connects to what they care about. Everyone should walk away understanding that technology can be and is used to solve real-world problems, and they should have hands-on experience doing just that in your program. It is likely that pre-existing curricula options will not perfectly or explicitly outline the social issue you want to explore; you will have to do the work to integrate the two.

But, how? We suggest developing ideas of various social issues, determining how you will communicate examples, and outlining how your lessons will connect to the social issues participants choose to focus on. In the beginning of your program, make time for participants to explore issues and determine what they would like to problem-solve. Choosing a theme, such as “education,” may be helpful to ensure that participants are exploring related issues and to better focus discussions and activities. From there, participants can make the issue their own, and choose something specific within education that they connect to and care about.
To determine your theme, ask yourself and your team: what social issues are common in our community? What issues are affecting the age group of our participants? How will the technology and skills the participants are learning connect to this social issue? Look at the examples below for some inspiration.

- Environment (climate change, pollution, natural resource depletion, weather, safety)
- Access & Ability (online resource accessibility, mental/emotional/physical disability awareness)
- Health & Medicine (affordable healthcare, mental health, substance abuse, eating disorders)
- Race & Culture (racism, whiteness, cultural appropriation)
- Gender Identity & Expression (socialization, toxic masculinity, toys, clothing, bathroom accessibility, menstruation/periods, gender reassignment, sexism)
- Agriculture & Food (nutrition, food growth and distribution, clean drinking water)
- Education (access, school-to-prison pipeline, discipline, learning resources, bullying, lunch/food options and access, evaluation, school sports, classroom space, funding, security, emergency preparation, music & arts)
- Local Community (gentrification, housing, vandalism and beautification, food sources and access, roads and highways, voting and voter laws, police and safety)
- Relationships (friendship, family, mentorship, classmates, teacher-student, significant others, partners)
<table>
<thead>
<tr>
<th>SOCIAL ISSUE</th>
<th>EXAMPLE OF PARTICIPANTS “OWNING IT”</th>
<th>TOOL/PLATFORM</th>
<th>APPLICATION EXAMPLES</th>
</tr>
</thead>
</table>
| Environment | **Participant 1:** cares about recycling in the community and wants to problem-solve how to make it easier  
**Participant 2:** cares about pollution and wants to problem-solve ways to reduce pollution | LEGO Robotics | **Participant 1:** designs robot to help community members recycle more efficiently  
**Participant 2:** designs robot that measures pollution levels |
| Education   | **Participant 1:** cares about bullying and educating kids about what bullying is  
**Participant 2:** cares about gender inclusion and having gender-inclusive bathrooms in school | Scratch | **Participant 1:** designs a PSA (Public Service Announcement) about bullying in school  
**Participant 2:** designs a PSA about gender-inclusive bathrooms |
| Animals     | **Participant 1:** cares about teaching kids how to take care of animals  
**Participant 2:** cares about eliminating animal cruelty | Codesters Game Design | **Participant 1:** designs an animal care simulator game  
**Participant 2:** designs game to find beauty companies that don’t test on animals |
CREATING AN ENGAGING LESSON

Armed with all of this information, you are now ready to map out the lesson you have adapted for your audience and your program. You will need to think about the materials you will use, any prep time that is necessary, and how you will engage your implementation team to make it happen. As you map out the lesson description, you may want to mark how much time you expect each portion to take, or create an agenda if you are running a day-long event. In the “Program Types” section on the Engage Students page, you will find example lessons and agendas that you can use for inspiration.

WANT TO DIVE DEEPER INTO LESSON DESIGN?

Consider NCWIT’s Engagement Practices Framework when designing your program content. The Engagement Practices Framework outlines actions facilitators can take to help broaden participation in computing and is organized around three evidence-based principles for engaging and retaining all students:

**Make it Matter:** All students are more motivated, perform better, and are more likely to persist when they can see how a lesson connects to their experiences, interests, goals, and values. And students who don’t fit the stereotype of someone pursuing computing may need even more explicit connections for them to envision themselves in the field.

**Build Student Confidence and Professional Identity:** Computing has come to be associated with some fairly strong stereotypes about who is a “computer scientist,” or more narrowly, a “programmer.” Anyone who doesn’t fit the stereotype may have difficulty seeing themselves in the field, and be less likely to have people supporting them in their pursuit of computing.

**Grow an Inclusive Community:** Students are more likely to persist when they have a community related to their academic pursuits.