



Material list:

- [You Can Be an App Developer](#) pdf
- Paper
- Something to write with – a pen, pencil, crayons or markers
- Optional: Post-it notes

* Note: You are going to start creating a prototype for a new app. You'll use the same supplies from step to the next.

Applications or apps are software programs that run on computers, tablets, or phones. App developers, or computer scientists, who create and build apps don't just think about the computer code that goes into building the app, they also come up with a design process well before they start writing the code. In this badge, we are going to look at how computer scientists use the design process to create new apps.

Step 1: Discover how apps can be used for good

Think about the apps you or your family members use on a regular basis. There are probably a lot of different ones, but if you look at them closer, you could probably group them together into different categories. Here are three common categories:

- **Entertainment:** people use these apps for fun; like games and video streaming apps.
- **Learn:** people use these apps to learn new things or skills; like translators and calculator apps.
- **Help:** people use these apps to solve problems; like map and calendar apps.

Can you think of an example of an app you or your family use that would fit into each of these categories? What need does that app fill? For example, if you use the Maps App, it will help you get directions for where you need to go. Or, your favorite game might fill the need to keep you entertained.

When app developers start working on a new app, they do not just think about a need the app is going to fill, they also think about who is going to use it. App developers call the person who is going to use their app the **user**. They may have to change the app depending on who is going to be using it, as the users' needs may be different.

For this activity you will do step 1 of the [You Can Be an App Developer](#) handout.

Step 2: Decompose the needs of your app user

Once app developers know the needs of their user and the problem or problems the user needs for the app to solve, they start thinking about how they'll make the app reach



that goal or fit those needs. App developers use something called decomposition to build their app. Step 2 of the [You Can Be an App Developer](#) handout; will help you think about how to use decomposition for your app.

Step 3: Design your app screens

Now that you know types of things, or features, your app needs to have to fit your user's needs, it's time to start building a prototype. Use step 3 of the [You Can Be an App Developer](#) handout to guide you in creating sketches for your app prototype.

Step 4: Create algorithms for your app that include events

Now that you have started to layout what your app will look like and you have created sketches for your different screens, it's time to think about how your user will get from one screen to the next and what will happen on each screen. Write the algorithms that will occur when an event happens on different screens. For examples, see step 4 of the [You Can Be an App Developer](#) handout.

Step 5: Share and improve your app with user feedback

Once an app designer has created a version or prototype of their app, they often ask their users to try it out to see if it fits their needs or solves their problem. Now that you have created your prototype, go back to your user and see if they have any suggestions to make it better or easier to use (user-friendly). Are there other things they would like to see included?

An important thing app developers look for at this step of the design process is to **debug** their app. A computer bug is when a programmer finds something in their code that doesn't work. When they fix the bug, they are said to be debugging. Often users provide feedback identifying bugs in the program. Did your user find any bugs you need to fix?

Now that you've earned this badge, you could give service by:

- Sharing with others what a "computer bug" is.
- Teaching others about breaking down a big problem into smaller parts in order to solve it.
- Sharing with others what I've learned about coding.

What are you inspired to do with your new knowledge?