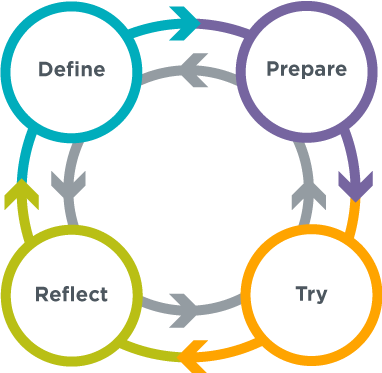
| **The Problem-Solving Process with Programming** |  |
| --- | --- |

# 

# Problem-Solving Process with Programming

The **Problem Solving Process** is useful when writing code for a challenging program. This process will help you clarify and break down a problem into manageable steps so you can easily identify the code you need to write for each step. 

## 🔎 Define

* Read the instructions carefully to make sure you understand the goals and intended outcomes of the program.
* If there is a sample program, run it to ensure you understand what it does.
* If there is starter code, read it to understand what it does.
* Rephrase the problem in your own words.
* Identify any skills you are being asked to apply.
* Decide which aspect of the problem you want to complete *first*.

## 📝 Prepare

* Brainstorm as many ideas for the different ways you could program the problem to solve all of its parts.
* Write out in plain English or pseudocode your ideas for how to solve the problem so that the program does what it is supposed to do.
* List what you already know how to do and what you don't yet.
* Describe your ideas to a classmate
* Review similar programs that you may have written in the past.
* Sketch out your idea on paper if needed

## 🖥️ Try

* Pick one idea from the previous step and document why you chose that idea.
* Implement your chosen idea and write one small piece of code at a time.
* Test your program often.
* Use comments to document and keep track of what your code does.
* Go back to the previous step to look at your other ideas for solving the problem if you get stuck
* Apply appropriate debugging strategies

**🤔 Reflect**

1. Compare your program to the defined problem … Were you successful in solving all aspects of the problem? How do you know?
2. Ask a classmate to try your program … Are there places where they struggle or show confusion?
3. Ask a classmate to read your code … Is your code and documentation clear and accurate?
4. Try to "break" your program … Did you find types of interaction or input that you could handle better?
5. Are there any changes or improvements you can make to this program or another like it in the future?