CS Discoveries Problem Solving Processes

Structured problem solving, through our Problem Solving Process, is a tool for student growth and development which has been woven throughout the entire course. While students learn the generalized Problem Solving Process in Unit 1, we have also developed versions of the process which more clearly articulate what each step looks like in the context of Programming, Design, and Data.

| **Unit** | **Problem Solving Process** |
| --- | --- |
| 1 | The Problem Solving Process |
| 2 | The Problem Solving Process for Programming |
| 3 | The Problem Solving Process for Programming |
| 4 | The Problem Solving Process for Design |
| 5 | The Problem Solving Process for Data |
| 6 | The Problem Solving Process for Programming |

|  | **The Problem Solving Process** |  |
| --- | --- | --- |

# The Problem Solving Process

Having a strategy for approaching problems can help you develop new insights and come up with new and better solutions. This is an iterative process that is broadly useful for solving all kinds of problems.

# Define

* Determine the problem are you trying to solve
* Identify your constraints
* Describe what success will look like

# Prepare

* Brainstorm / research possible solutions
* Compare pros and cons
* Make a plan

# Try

* Put your plan into action

# Reflect

* Compare your results to the goals you set while defining the problem
* Decide what you can learn from this or do better next time
* Identify any new problems you have discovered

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

## 

# Define

* Read the instructions carefully to ensure you understand the goals
* Rephrase the problem in your own words
* Identify any new skills you are being asked to apply
* Look for other problems you’ve solved that are similar to this one
* If there is starter code, read it to understand what it does

# Prepare

* Write out an idea in plain English or pseudocode
* Sketch out your idea on paper
* List what you already know how to do and what you don’t yet
* Describe your idea to a classmate
* Review similar programs that you’ve written in the past

# Try

* Write one small piece at a time
* Test your program often
* Use comments to document what your code does
* Apply appropriate debugging strategies
* Go back to previous steps if you get stuck or don’t know whether you’ve solved the problem

# Reflect

* Compare your finished program to the defined problem to make sure you’ve solved all aspects of the problem
* Ask a classmate to try your program and note places where they struggle or exhibit confusion
* Ask a classmate to read your code to make sure that your documentation is clear and accurate
* Try to “break” your program to find types of interactions or input that you could handle better
* Identify a few incremental changes that you could make in the next iteration

|  | **The Problem Solving Process for Design** |  |
| --- | --- | --- |

# Define

* Identify potential users
* Interview users
* Read user profiles
* Identify needs and wants

# Prepare

* Connect needs and wants to specific problems
* Research how others have addressed these issues
* Brainstorm potential solutions
* Discuss pros and cons
* Identify the minimum work need to test your assumptions

# Try

* Draw your product on paper
* Develop a low fidelity prototype to communicate your design
* Share prototypes with potential end users for feedback

# Reflect

* Present to stakeholders
* Review user feedback

|  | **The Problem Solving Process for Data** |  |
| --- | --- | --- |

# Define

* Decide what problem you are trying to solve or what question you are trying to answer
* Make sure you understand your target audience (it could be you!) and what specifically it needs
* Identify the parts of your problem you could address with data, and how more information could help

# Prepare

* Decide what kinds of data you will collect
* Decide how you will collect the data and in which format you will collect it
* Anticipate possible challenges in data collection and change your plan to account for them
* Develop a plan for how you will analyze your data and make sure your data will be useful for that kind of analysis

# Try

* Collect your data using the plan you created
* Clean your data by removing errors, unexpected values, and inconsistencies
* Visualize the data by creating tables, graphs, or charts that help you see broad trends in your data
* Interpret the trends and patterns in your visualizations based on your knowledge of the problem

# Reflect

* Review what you’ve learned about your question or problem
* Decide if what you’ve learned has solved your problem and allows you to make a decision, or if you’ll need to go back to one of the previous steps