

Computing with a Purpose

Lesson 4.1: Innovation of Computational Problem Solving

Preface

The final lesson of the course allows you to apply all that you have learned in a student-defined, student-driven development. Whether creating an app, a website, or a physical computing device, you will apply computational thinking practices and a strategic development process to create computational artifacts that solve problems and create value for others. You will collaborate the way computing professionals do as you pursue solutions to authentic needs.

Create Performance Task: Your Development Process

Essential Questions

1. What was your role on the Scrum development team?
2. What is the purpose of your program?
3. Where does your program integrate mathematical and/or logical concepts?
4. What does one of the algorithms in your program do?
5. How does an abstraction you created manage complexity in the program?
6. What part of the code did you develop?

Resources

[Interpreted Performance Guide](#)

[Response Template: Create Performance Tasks](#)

Problem Introduction

Many times computing professionals are presented with a problem and asked to develop a solution. You have experienced this throughout this course as you were given projects and problems with specific constraints to develop solutions to. Your projects are sometimes narrow in focus, so that you acquire the technical skills needed to tackle more open-ended problems. Even the open-ended problems at the end of each unit still contain some constraints for you and your team.

What if you were given the opportunity to pursue your own idea? a problem or opportunity you are interested in—not one defined by someone else? In this final development process, you will model processes that will help you identify and narrow possible ideas that are open-ended and can be done within the time and constraints you have in this class.

Where Do I Start?

The following steps describe the process in general terms:

- Brainstorm: Find an Idea to Pursue
- Document Your Development Milestones
- Prepare, Investigate, and Plan
- Design, Create, and Test
- Evaluate and Reflect
- Present

An open-ended design and development process can sometimes be challenging because you are accustomed to having many things in a project defined for you. In Unit 4, it is time for you to apply what you know to a topic that is important to you.

The best place to start is by brainstorming as many possible problems that you can think of. The more the ideas and the earlier you start, the better. You can begin by exploring what you know or problems you have come across, look into what other people talk about as problems in the world today, or search a variety of sources that identify opportunities. The key is to brainstorm a long list and start early.

In preparation for your final course problem, you should begin early in the year applying strategies and documenting problems that might be worth pursuing a solution to. Keys to the process are:

- Ensuring that the solution or what is created has value to others
- Decomposing the problem
- Incremental and iterative development and testing
- Collaborative strategies and diverse perspectives

Problem Timeline

Day 1	<ul style="list-style-type: none"> • Introduction and overview of teacher defined requirements • Preview evaluation criteria - Interpreted Performance Guide • Preview example(s)
Days 2–4	Part A: Find an Idea to Pursue <ul style="list-style-type: none"> • Identify ideas for a product that could be used by people other than the development team.
Day 5	Part B: Document Your Development Milestones <ul style="list-style-type: none"> • Video submission requirements • Individually written response requirements • Program code
Days 6–8	Part C: Prepare, Investigate, and Plan <ul style="list-style-type: none"> • What does “collaboration” mean? • Independent program development requirements • Scrum Planning <ul style="list-style-type: none"> ◦ Define Roles ◦ Collect User Stories ◦ Initial Product Backlog Meeting ◦ Sprint Planning Meeting ◦ Sprint Retrospective ◦ Groom the Backlog ◦ Next Sprint ◦ Final Sprint ◦ Development Milestone Retrospective
Days 9–23	Part D: Design, Create, and Test <ul style="list-style-type: none"> • Version control • Testing and refining
Days	Part E: Evaluate and Reflect

24–32	<ul style="list-style-type: none"> • Development milestone retrospective
Days 33–35	Part F: Present



PLTW DEVELOPER’S JOURNAL Document all problem work in your PLTW Developer’s Journal.

The Development Process

Part A: Find an Idea to Pursue

Part B: Document Your Development Milestones

Part C: Prepare, Investigate, and Plan

Part D: Design, Create, and Test

Part E: Evaluate and Reflect

Part F: Present

Find an Idea to Pursue

Problem-solving Categories

Below is a list of several categories that might help you start to define a problem worth pursuing a solution to. Regardless of what you choose to pursue, you will need to provide evidence to your teacher that what you are proposing to create has value for others.

1. **Personal Experiences or Interests**

- What do you like to do? Complete the sentence:
 - “Don’t you hate it when...?”
 - “I love animals...”
 - “I play soccer and could use...”

2. **School Culture**

- Students sometimes feel disconnected from their school. What do you see every day here at school that could be done better?

3. **Education**

- How can you make learning more fun, easier to understand, or just better? What can help students learn for future years?

4. **Technical Problems**

- People often complain about technical problems and want a different way to do something. What do they complain about?

5. **Health, Wellness, and Safety Issues**

- People are often hurt or killed because of a health, wellness, or safety problem. What causes injuries or health issues?

6. **Community Service and Volunteering**

- Schools and communities support each other by coming to local games and using local services for celebrations. Are there problems that you can help solve to strengthen those school and community ties? What is a problem in your community?

7. **Environment**

- Recycling, protecting natural resources, and limiting use of non-renewable resources are some of the basic environmental challenges that exist. What are some environmental challenges that exist where you live?

8. **Emergency Preparedness**

- Disaster communication and risk tracking are common problems communities look to solve. What should people do in emergencies? What are the potential risks of a specific emergency?

9. **Legal Issues**

- What problems cause people to take each other to court?

10. Global Challenges

- Global challenges, such as poverty, are well-documented problems, but very challenging to offer an exact one-size-fits-all solution for. What are some global challenges no one has solved that you care about?

11. Other

- What game or other creative expression idea are you interested in? Remember that you still need to provide evidence to your teacher that your idea has value to others.

Problem Identification

There are essentially two ways you can identify a problem.

- Find an opportunity based on your own life experiences.
- Find an opportunity based on observations of what others have identified as a problem that needs to be solved.

A good place to start is with what you know or problems you are interested in.

1. Create an interest inventory:

- What are your hobbies, passions, your interests?
- What are you an expert at?
- What do you want to learn more about?
- Have you observed things that bug you?
- Which of the Problem-solving Categories did you write the most for?
- Which did you find the hardest to move on from or change your focus from?

Research

What are some obvious problems in our society today? You can start by doing a quick search of “global challenges”. The tough part about trying to tackle global challenges is that you most likely will not come up with a solution for many reasons, but that does not mean you should not try. If global challenges are too daunting or not of interest to you, try exploring the local news to identify local challenges (pet populations, recycling, traffic).

Personal Observation and Experience

What problems have you come across in your own experiences or that other people have identified? On your own, begin exploring potential problems you or other people outside your classroom are encountering that you might be able to create a computational solution for. Think creatively about their needs and ideas that you are interested in.

Problem-Solving or Creative Expression?

Whereas solving well-defined problems for people is the most noble goal in computer science, sometimes developers create computational artifacts to meet a need that no one has even defined yet. You may choose to select an idea that allows for creative expression and might entertain

others.

Regardless of whether you are solving a problem or creating a software program as artistic expression, it is important that you can demonstrate that what you will create has value to others.

Primary vs. Secondary Research

2. Collect and add all evidence of research for future reflection and review.

While doing your research, be sure to document where you get the information. This way, if you need more information later, you can return to retrieve that additional information. Consider having a page in your journal to track sources or an electronic place that you can share with the teacher and other team members.

While researching, ask yourself the following questions to verify whether the information is from a trustworthy source. Is it a primary source or a secondary source? Does the URL support a reliable source?

Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

You will also need to ask yourself whether the URL domain is reliable. The following tips will help you navigate URL domains in your research:

- .gov sites have a responsibility to provide information that is as accurate as possible.
- A .org domain name can be bought by any organization. Consider whether that organization is an authority on the subject. An organization that supports blimp travel may provide biased or inaccurate information about other ways of flight travel, to support their own cause.
- .com sites can be purchased by anyone and there is no regulation on what can be on those sites.
- .edu sites can be used by knowledgeable professors or by their students who are trying to turn in and post an assignment the night before it is due. Consider the source publishing the information. Do they have the background experience for you to consider them trustworthy on that subject?

Brainstorm Possible Problems

3. List as many potential problems as you can to pursue for a potential solution. In this brainstorming exercise, there are a few guidelines:
 - A list of 10 potential problems is a good place to begin. The more problems you identify, the better off you will be later in the process. For each potential problem or opportunity, identify a stakeholder who could provide insight or feedback.
 - Use your brainstorming on the 10 categories.
 - Do not offer solutions yet. Make sure you are identifying problems. Do not offer solutions

you are interested in creating.

- Write in complete sentences. Could someone pick up your journal and understand the problem you are describing?
- Pay attention to your constraints. While brainstorming, your thoughts should be a “free flow” of ideas. Do give some consideration to how much time you have, the resources you have access to, and the capabilities of your team. If a large idea is presented, brainstorm on smaller parts that may be done within the timeframe that you could add to in the future.
- Your instructor will help make determinations on what may be accomplished in this course.

Prioritize Opportunities: Who Are the Stakeholders?

4. After you have created a list, identify five problems or opportunities you feel have the best potential for developing a product around. Pay careful consideration to the opportunity and availability to actually tackle the development process. (Time is typically your biggest constraint.)
5. Prioritize your list from the ones you are most interested in pursuing to ones you are least interested in. Mark each one for your potential ability to complete an attempt at the solution. Be as descriptive as possible in describing the problem.
6. For each of these potential ideas, provide a rationale for how it will have value to the stakeholder you identified.

Present Your Top Ideas

7. In less than three minutes, you will report on your top two ideas. Each potential idea should describe:
 - What is the problem or opportunity?
 - Who are the stakeholders?
 - Provide a rationale for how this idea could provide value to the stakeholder.
 - What is the initial user story?
 - Is the source credible and reliable?
 - Are there other potential users?

Document Your Development Milestones

Resources

Scrum

After you have found an idea to pursue, you will form teams and begin to to define how you will document your development process.

Development Milestones

There are moments in any user-centered and iterative process where you need to share information and documentation with a wide variety of audiences. Within your development process, you should continuously be documenting and refining:

- What is the purpose of the program you are creating?
- Who would use this product?
- What user stories have you captured from the people using the product?
- Does what you are creating have value to others?
- What are the roles on your team and how will your effectiveness be evaluated?
- What parts did you create?
- What challenges did you face and how did you overcome them?
- What are abstraction in my program and how are the managing complexity?
- What are algorithms doing in my program?
- How will we manage files and version control?
- How will we share what is completed in a sprint?
- How will we communicate the entire story of our development process and not just the product that was created?
- What are the best practices of a code showcase or code review?

Preview the Interpreted Performance Guide and Final Presentation Expectations

The best way to determine how and what you will document is to preview the end of unit expectations. Some are defined in the [Interpreted Performance Guide](#) and some are defined by your teacher. You teacher is the best source for determining where you should document things and how you will communicate your development milestones.

Present

Form a Team

1. Identify others in your class who have similar interests or who presented new ideas that interest you.
2. Based on the potential ideas that you and your classmates described, your teacher will help direct the formation of teams.

Suggestions

The original owner of the idea has final say on who will be on the development team.

A team may wish to approach another team and ask that team if they might pursue an idea they identified but are not pursuing.

3. When you have the appropriate number of people on your team, use the following guidelines to begin to plan your development milestones.
 - Describe the problem in clear, easy to understand words that form a summary in complete sentences. Anyone who reads the statement should understand your problem or idea.
 - Who is the intended audience or end user?
 - What is the initial user story?
 - Is the source credible and reliable?
 - Are there other potential users?
 - Explain why you chose this idea over other idea proposals.
 - Define the constraints of the problem. What does the solution need to do?
 - What can reasonably be accomplished in the time frame for this development process?
 - Show your development outline to your teacher to get approval.
4. Make sure that all team members have written explanations of the teams development milestones.
 - It is important to verify team alignment of goals now before people spend time developing something that does not meet the development milestones or final product goals.

Reminder: Your journal should reflect all the research you did as an individual and the research that will guide your team moving forward around the topic you have all chosen to pursue.

Team Responsibilities

Roles on a Scrum Team

The **Product Owner** is the keeper of the requirements. They:

- Provide the “single source of truth” for the Team regarding requirements and their planned order of implementation.
- Work closely with the team to define the user-facing and technical requirements, to document the requirements as needed, and to determine the order of their implementation.
- Maintain the Product Backlog (which is the repository for all of this information), keeping it up to date and at the level of detail and quality the Team requires.

The **Team** is a self-organizing and cross-functional group of people who do the hands-on work of developing and testing the product. Teams usually consist of four students, including the product owner. The team:

- Must have the authority to make decisions about how to perform the work.
- Must decide how to break work into tasks, and how to allocate tasks to individuals, throughout the Sprint.

It is important to note that while team members have some specific roles, all team members must be involved in the creation of the final product. When the problem is decomposed, some features or backlog items might even be created using pair programming. Also, all team members should be ready to report daily on the progress made and what they will do next.

The **Scrum Master** is responsible for making the whole development process run smoothly, for removing obstacles that impact productivity, and for organizing and facilitating the critical meetings. The Scrum Master's responsibilities include:

- Remove barriers between the development team and the product owner so that the product owner directly drives development process.
- Facilitate creativity and empowerment.
- Improve the productivity of the development team in any way possible.
- Make sure each increment of functionality is potentially usable.
- Keep information about the Team's progress up to date and visible to all parties.
- Assess responses and feedback.

The Product Owner and Scrum Master will help make final decisions for the group after hearing different issues and sides of disagreements about the development process as they arise.

Each team member is responsible for providing access to files for the whole team and capturing specific milestones within the project.

Set Team Expectations

As a team, you will either succeed together in meeting your goals, or you will not. Success is not the responsibility of any one individual. While you have defined roles to keep the development process progressing, it is the responsibility of everyone on the team to do their own part and hold each other accountable for their parts.

5. Establish team expectations:

- How often will the team members report on their progress to each other?
- How will team members show evidence of progress with the teacher when the teacher selects a random person to provide an update on the team's progress?
- How will the team handle conflicts?

You have team leadership roles, but what process do you expect of your leader and of yourselves as team members? Outline some qualities of a leader for talking through, listening to, and resolving conflicts between members of the group. If the conflict is between a team lead and a team member, how will that conflict be handled? There is little time in the development process to get upset, hold grudges, or make other product choices based on emotional responses, so how will you handle these events?

- What plan is in place for an unexpected absence of a team member to ensure development milestones can continue?

- Will you download the product parts at the end of each class period? Will you record everything in your PLTW Developer's Journal so that if you are gone, others know what you were working toward? Will you have a collaborative space online that allows you to collaborate in real time and track what has been done and needs to be done?

Contact a Stakeholder

6. As a team, define a strategy to contact a stakeholder to capture their insights and continue to develop the user story about the topic your team selected.

Prepare, Investigate, and Plan

Resources

Scrum Poker

Create a Backlog

1. Further develop your idea for your proposed solution. Create a backlog listing portions of the solution to be developed. Put the list roughly in the order in which you would develop the solution.

Note: It can sometimes be difficult to determine which user story from your backlog to approach next. Scrum poker is one tactic to help a team determine where to start.

Scrum Poker

2. Each team member should be honest about their strengths and weaknesses to help divide the backlog into milestones to be accomplished in the time frame of the development process. Determine who on the team will develop which parts.
3. To make sure you get a working solution sample before the end of the problem time frame, make the first few portions listed in the backlog relatively smaller steps than other portions. You may not be able to complete your entire backlog, but picking some steps will help you have a working solution to show, even if it is not complete.
4. Each person should decide which portion of their backlog to develop first and break it into several sprint tasks to be approved by the group.
5. Based on everyone's needs, develop a design and record it in a shared location. Having a set design with consistent naming of design elements will help minimize time lost when bringing everyone's work together.
6. Decide on how you will do version control with naming and sharing files to make sure everyone has access to all the files in the event that someone is absent for a few days. Iteration control should consider:
 - How will you store old file versions? Sometimes parts of a program work individually, but when they are put together, they do not. This can be from conflict in names, multiple event handlers for the same event, or other issues. It is important to have backup files in the event that a later version stops working and you need to restore to a previous working version.
 - How will you name files with each iteration? If each team member is working on a different file, where each person saves that file at the end of each period, and you are collecting them all together, it is important to distinguish what each file is at a quick glance. Having specific names that identify what the file does, who wrote it, and when they wrote it will help with finding files in an ever-growing list of files. Do not rely on the

created or modified dates on files. If you just open a project, it will show an updated modified file. Just because a file was created before the others does not mean it does not have the most recent version of the work. Have a plan in place to take out the guesswork, as you do not have time to waste in opening a bunch of files to find the correct one.

- How will you make sure that all team members will have access to your work before you leave the classroom each day? The development process window is short. One person who gets the flu can miss most of the problem development process. Expecting people to work while they are out sick is not a solution. You need to have access to the files and the documentation that guides each other's work.
7. Review guidelines for collaboration and using the platform you are developing in to identify any potential difficulties developing as a team.
 8. Review the [Interpreted Performance Guide](#) that will be used to evaluate your team's development milestones at the end of the problem time frame. Based on your plan, what level of performance are you headed toward? What kind of adjustments can you make to get full marks and have a successful development product?
 9. Make sure your teacher approves your plan for development milestones. The timeline is short, so you will not have enough time in class to redesign and redevelop parts of your product and still meet all the development milestones.
 10. You should be able to communicate to your teacher:
 - ☐ Your role on the team.
 - ☐ What your team is working toward.
 - ☐ Who your stakeholders are.
 - ☐ How you will collaborate.
 - ☐ What your team expectations are for conflict.
 - ☐ How you will handle iteration control.
 - ☐ What you will begin working on the next time you walk into the classroom.

Scrum Poker

When you are developing using the Agile methodology, it can sometimes be difficult to determine which user story from your **backlog** to approach next. **Scrum poker** is one tactic to help a team determine where to start.

To play scrum poker, all you need is a set of cards with values 1–10. A standard deck of playing cards will work, counting the Ace as value 1.

For each user story, each team member privately selects a value between 1 and 10 to rank that user story, first in terms of difficulty to code, then in terms of benefit to the end user.

- With regard to difficulty to code, 1 represents very easy, and 10 represents very difficult.
- With regard to benefit to the end user, 1 represents minimal benefit, and 10 represents maximum benefit.

Once each team member has selected both values for a given user story, they all reveal their cards simultaneously and record the results before moving on to the next user story.

When all user stories have been scored on difficulty to code and benefit to the end user, the team considers all values together to determine the order in which to prioritize the backlog.

Once the backlog is ordered, the team creates a solution to the user story that they decided had the highest priority.

If you do not have access to cards, you may use this page to make your selections on a tablet.

Difficulty to develop

Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

Design, Create, and Test

Iteration and User-centered Research

You and your team will apply all that you have learned to solve a problem or meet a need. As you work with your team, it is important to address the critical parts of any development process.

- Are we seeking information from stakeholders in the problem?
- Are we listening to the end user and using their input to shape what is created?
- Are diverse perspectives represented? Will any solution meet the needs of all? How can we make sure?
- Are all team members contributing effectively?

Each day your team should run through a quick update and check in. Below are suggested touchstones for your team to discuss in update and check-in meetings.

Daily Check

1. What did I accomplish yesterday?
2. What am I trying to accomplish today?
3. How does what I am doing fit with the work of my team members?

Reflect on Iteration

1. How is the team storing old file versions?
2. How is the team naming files with each iteration?
3. How do I make sure that all team members have access to my work before I leave the classroom today?

Reflect on Scrum Development Milestones

1. Is what I am working on helping to meet the stakeholders needs, wants, and goals?
2. After I finish this part, what will I work on next to meet the stakeholders' needs, wants, and goals?
3. If I am asked to describe the progress of my team, am I prepared to report on what everyone is doing and how it is connected to a specific user story?
4. How am I collaborating and getting input from others on my team?
5. Is the feedback I am providing to my team being well received? If not, how can I adjust my feedback to help the overall development product be successful?

Thinking Ahead

Before you complete this part of the problem development process, practice the final presentation of

the idea and development milestones as a team. You should know what parts you will be speaking about so you can practice and review before your presentation. Your team should have a method in place to gain feedback from your peers and stakeholders, to continue to adapt your development product so it appeals to people outside your team.

Evaluate and Reflect

Code Showcase

As you near the completion of your development process, share what you have created and get feedback from as many users as possible to evaluate the effectiveness of the project.

Presentation Prep

- ☐ My team has practiced the presentation.
- ☐ I know what part I'm going to speak about.
- ☐ I know the cue that means it is my turn to present.
- ☐ I can show evidence of the process and current state of development product.
- ☐ I can answer questions from others.
- ☐ I know how my team will collect the feedback to address before the final presentation.

Evaluating Feedback

- All feedback is important. If someone took the time to tell you they ran into an issue, did not understand something, or did not like something, consider what your team can do to address those issues.
- Are there trends in the feedback that point to some issues that the majority of users experienced in your practice presentations?
- Some feedback is more important to address than others. Prioritize what feedback you want to adjust for first, over what may not be possible during the time you have to revise.

Acting on Feedback

- Divide the tasks to be completed.
- Implement the adaptations to the development product.
- Reflect on what went well. Then reflect on ways to improve, so the next presentation goes better.

Thinking Ahead

As you work toward the final presentation, ask whether you and your team can address the following development goals:

1. Can I describe the stated purpose of the program?
2. Can I describe difficulties and/or opportunities I encountered and how I dealt with them?
3. Can I describe two algorithms that work independently and in combination in my program?
4. Can I describe an abstraction that helps manage complexity in my program?

Present

Evaluation Criteria

As directed by your teacher, develop the following items to turn in with your final presentation. Each of these evaluation criteria are defined by The College Board for create performance tasks in computer science. You will find them referenced in the Interpreted Performance Guide.

1. A copy of your Developed Program Code (DPC).
2. A Video of the program code running (V).
3. Independently Written Responses (IWR) describing your development process and product; each team member does their own.
4. A presentation to your class.

Developed Program Code (DPC)

The purpose of the DPC response is to demonstrate that you can:

- Use appropriate citation of program code. Any program code that you did not write should be cited, and credit should be given to the author.
- Provide commented code. If the programming environment allows it, capturing comments in the programming environment is preferred. If the programming environment does not have this type of functionality, you can include comments with the capture of the program code for submission.
- Identify an algorithm you created (for ease of identification, use an oval to represent the algorithm).
- Identify an abstraction you created (for ease of identification, use a rectangle).

Video of Program Code Running (V)

Your teacher will review expectations for how to capture and share the video. The purpose of the video is to show that you can:

- Demonstrate the functionality of your code.
- Use accurate terminology to describe and explain your code.

Independently Written Response (IWR)

The purpose of the IWR is to demonstrate that you can:

- Describe the purpose of your developed program code.
- Use several mathematical and logical concepts in your code.
- Implement algorithms that work independently and together.

- Develop and use abstractions to manage complexity (procedures, abstractions provided by the programming language, APIs).
- Describe difficulties and/or opportunities encountered and how they were dealt with.

Classroom Presentation

The purpose of the Classroom Presentation is to show that you can:

- Describe the story of your development process, milestones, and product from conception to evaluation.
- Describe how your collaboration was organized and what roles each of your team members played.
- Describe how diverse perspectives played a role in making your development product better.
- Receive and implement feedback from others to improve the effectiveness and value of the development product.