How to Complete a Mathematics or Computer Science Project

What is a computer science project?
A computer science project uses coding language to develop information processes or programs to demonstrate, analyze, or control a process/solution. Sometimes robots or intelligent machines are used to use the coding language and perform tasks.

What is a mathematics project?
A mathematics project involves using math to describe relationships between things. It could describe the relationship between the amount of force applied and the distance a ball travels, or it could be the probability of a specific event happening. The relationship is described as an equation, formula, or algorithm.

1. Get an Idea for Your Mathematics or Computer Science Project
Like a science fair project, a mathematics or computer science project starts with a problem, but the problem is a bit different. In science, you might be asking a "What if?" question, such as "What will happen if I add food coloring to saltwater before I evaporate the water?" Computer science and mathematics, on the other hand, look at the real world, see a problem, and use a coding language or equation to try to solve the problem. Examples in coding could include developing an application, designing a game, writing a program for a robot, or programming a microcontroller (Raspberry Pi, Arduino, AdaFruit Circuit). Examples in mathematics could include analyzing probability, calculating angles, devising an algorithm to solve a problem, or game theory.

There are three categories in the Mathematics and Computer Science division of the Science and Engineering Fair.

- ***Coding*** focuses on the study or development of software, information processes or methodologies to demonstrate, analyze, or control a process or solution. Learning to code could lead to a career in many fields, including:
  - video game developer
  - cybersecurity
  - programming languages
  - operating systems
  - application development
- ***Mathematics*** addresses studies of the measurement, properties, and relationships of quantities and sets using numbers and symbols. This includes the study of numbers, geometry, probability, and statistics. Career fields that use mathematics include:
  - financial analyst
  - computer systems analyst
  - statistician
  - game theory specialist
  - geometry and topography
  - mathematician

- ***Robotics and Intelligent Machines*** projects use machine intelligence to complete a task or reduce the reliance on human intervention. If you have an interest in computer science, you might look at a career in:
    - biomechanics
    - cognitive systems (artificial intelligence)
    - robot kinematics (how robots move)

2. Start Mathematician's or Programmer's Log Book

A detailed log book with accurate records allows programmers and mathematicians to describe their coding or mathematical processes and reflections on algorithm development and debugging so others can follow the process. Your log should be a bound notebook (such as a composition book). It should be done fully in ink. That's because it can be used as a "legal document" to prove your code/equation is your creation. In the real world, the log book is used as proof for patents and copyright. It can even be used as evidence in lawsuits over who was the first person to come up with a new idea. That's a pretty powerful book!

Don't worry about making mistakes or making a messy drawing. Mistakes are part of the process of learning and discovering. If you make a mistake, just draw one line through the mistake and keep going. Don't tear out pages or scribble out anything. It's possible that a string of code or equation you thought wouldn't work early in the process turns out to be the solution to your problem.

Setting Up Your Mathematician's or Programmer's Log Book

- Your log book is slightly different from one used by a scientist or engineer. For this division, your log book is a daily record of your progress as you work toward meeting your goal. It is very important that you record detailed notes about the work you complete on your project each day. Each day's entry will include **Daily Work** and **Daily Reflection**. Be sure to label each part for every entry that you make.
- If you make a mistake, draw a line through it and rewrite it. Do not erase or white out a mistake.
- In the **Daily Work** part of each entry, write down all the things you do or plan concerning your project each day. ***Make sure you date every entry.*** Think of it as a blog post each day:
    - What did you do today for your project?
    - Did you record any ideas for your program or equation (sketches of characters, tasks for your robot, story ideas for your game, input/output for your microcontroller, calculations for your problem)?
    - Did you change any of your code or equation today? Did you take screenshots before and after you made changes to your code? Did you rework your calculations to check for mistakes?
    - Who did you talk to about your project?
    - What did you research? Make sure to add the bibliography information for each source.
    - Give details! Each day's entry should show the progress on your project.

- In the **Daily Reflection** part of each entry, think about what you learned today:
    - What roadblocks or obstacles did you run into today? What new ideas or questions have come about as a result of working through the roadblock or obstacle?
    - What resources did you use to solve your problem (tutorials, asking a teacher for help, looking up code)?
    - If you made changes to your code or equation, what did you learn from it? How will your new learning help you be successful next time?
    - What successes did you have today? Did your successes spark new ideas for your code/program/equation?
    - Do you notice any patterns, repeated structures, or trends?

3. Complete the Project Approval Form - 2 pages

This form lets your teacher know what you've chosen for your project. It gives an overview of your project with enough detail that anyone who reads it can get a pretty good idea of what you will be doing. Once your teacher approves the project, he/she will give this form back to you. It will have a list of other forms you will need to complete before you begin your project. ***Make sure you keep this signed form and all forms you complete--they are required to be turned in with your project.***

4. Become an Expert in Your Problem

The research phase of your project is very important. This is where you learn everything you can about the topic of your project. If you are trying to solve a problem, you need to understand the problem. Spend some time getting background information. Good research will help you become an expert on your topic. Remember to write down the bibliographic information about each source you read, consulted, or tried to contact. Some ideas of places to go for research are:

- library
- internet--Make sure it is a ***reliable*** source of information (talk to your school media specialist about this).
- experts in the field
- Write to companies involved in your field.

5. Complete Ethics Agreement and Risk Analysis and Designated Supervisor Form

By signing the *Ethics Agreement*, you are saying that you won't copy someone else's work. You can refer to someone else's work, but you have to cite it in your log book and on the bibliography. Copy-and-pasting images, words, etc., from the internet is considered plagiarism. If you identify *where* you got each part of what you copied (cite the source), you have done your job.

The *Risk Analysis and Designated Supervisor Form* is used to state all the risks in your project. Risks might include:

- the materials and programs you are using. How can you stay safe when you use them?
- the location you are testing in. Is it close to a road or body of water?
- the tools you may use if building a robot or other intelligent machine.

In this handbook, the Risk Assessment and Safety Considerations section will help you complete this form.

6. State the Problem in a Question Form

Your project problem is how you will develop a program using a coding language/equation/formula to solve a problem. The problem should be a practical need. For coding or robotics problems, are you coding a completely new program or are you modifying (changing) existing code to make it work better in certain conditions? For mathematics problems, are you trying to find the probability of a specific outcome? Are you trying to find an equation that will describe a relationship between two outcomes? Whatever it is you are trying to do, your final program should be a process/solution to the problem you identified. Your problem should also be very specific. For instance, if you want to design a game, be very specific about which coding language and tasks your program will perform. For example, you might ask, “How can I use Scratch to design a chase style game?” For mathematics, you might ask, “What is the probability of winning a Rock, Paper, Scissors game?”

7. Research

Mathematicians and computer scientists need to get a full picture of the problem they are addressing before they start developing their programs/equations.That's where research comes in. For example, if you are programming a robot, find out the coding languages that are compatible with that robot. If you are using a microcontroller to program circuits, research what you will need to build the circuits, how the parts of the microcontroller operate, and the most efficient coding language for the microcontroller. If you are devising an algorithm for solving a Rubik’s Cube, research three-dimensional puzzles, how a Rubik’s Cube moves, and solutions for solving Rubik’s Cubes. Research helps you to fully understand the problem and possible solutions before you start your design.

For the Science and Engineering Fair, at least ***3 sources*** are required for the research phase. These sources must be documented in both the log book and on a bibliography. Interviewing a computer programmer, statistician or other expert in the field of your project is an acceptable source.

8. Develop a Project Goal

Your project goal should start as a brainstorm of several solutions/processes to your problem. Don’t stop at just one. Brainstorm alternative solutions/processes that might solve the problem, then choose the one that you think best fits the specifics of your project goal. At judging time, you will be asked about different ideas you brainstormed and why you thought your solution/process was the best. All of your solutions/processes should be in your log book, with detailed labels and components of your program. Programmers might include designs of a maze a robot navigates through, sketches of a character they are developing for Scratch, illustrations of circuits, or a display menu for an application. Mathematicians might include sketches of two-dimensional or three-dimensional objects, possible formulas, or diagrams of angles.

9. List Materials and Programs

Include any materials you plan to use, including specific robots, devices, and materials you need to complete tasks when appropriate (tape, construction paper, batteries, sensors, wire, LED lights, tape measure, protractor, compass, etc.). Also include a list of the programs and coding language (Scratch, Arduino IDE, MakeCode, Tynker, JavaScript, HTML5, C++, etc.), or formulas you will need.

10. Write an Algorithm or Testing Procedure

An **algorithm** is a to-do list for a computer or mathematics problem. A recipe is a good example of an **algorithm** because it tells you what you need to do step-by-step. It takes inputs (ingredients) and produces an output (the completed dish). The algorithm is your procedure for developing your code. Using statements, write the steps you will need to code to perform the tasks for your solution/processes. Your algorithm could be written as an outline, a list of steps, in a flow map, or in a storyboard.

A **testing procedure** is where you design a way to gather data about your mathematical problem. If you are devising a new way to solve a Rubik's cube, how would you test your formula?

11. Develop - Test - Modify

Mathematics: First, develop a formula, ratio, or solution for your problem. Test it according to your Procedure. The next step is analyzing your results and creating a graph. Was there a particular problem with one part of your formula/solution? Did you have any outliers in your data that need to be considered? After you test it several times, do you see any trends in your data tables/graphs? Adjust your formula, ratio, or solution and test it again. Keep repeating this cycle until you have a solution that best meets your needs.

On your Mathematics board, you will be required to display your process for developing your formula, ratio, or solution. Projects should include your testing procedure, graphs of data, trends and outliers, and your reflection. **If you used a mathematical application on your computer, you may choose to bring a computer to judging. *No computers will be supplied at the regional fair.***

Coding/Robotics and Intelligent Machines: Using your algorithm (step-by-step procedure), write your code to perform the tasks of your project goal. Good programmers run their programs after they write each line of code. They are testing that the code runs correctly. If an error in the code is discovered, it is easier to find the error in the string of code when you test your program frequently. Finding and correcting errors in your program is called **debugging**.

On your Coding or Robotics and Intelligent Machines board, you will be required to display changes you have made as you develop your program. Screenshots will help you document these changes. Projects should include screenshots of your initial program, several changes as you debug and modify your code, and your final program. You also might want to take screenshots of strings of code that you feel are significant to your project goal, a complicated design, or were challenging to develop. **Judges have to be able to inspect your final code. A printed copy of your entire final code is recommended. Student programmers may also choose to bring a computer to judging for this purpose. *No computers will be supplied at the regional fair.***

12. Final Reflection

Your final reflection should demonstrate your thinking about what you have learned. You could create a timeline with descriptions or steps in the process that show the creation of your project from start to finish. Discuss lessons learned from your project, including ideas you have for future research, steps or processes you would do differently, and other lessons you have learned that will help you with your program next time.

- Tell the story of your project. Why did you choose to solve this problem?
- Describe the design process you used in your project. Imagine you are telling a classmate about it who is interested in working on a similar project.
- Tell the story of how you solved roadblocks or obstacles that came up in the development of your program/equation. Give examples of a few challenges you encountered and how they created problems in your code.
- What resources did you use to problem solve?
- How did the ideas for parts of your program/equation change after you started?
- What new ideas, questions or goals have come about as a result of your work on this project?

13. Communicate Your Results/Construct a Display

Mathematicians and computer scientists share their findings with others. If your program/equation solves a problem, it is good to let others know about it! You should be able to fully explain all parts of your project: How did you come up with the problem?

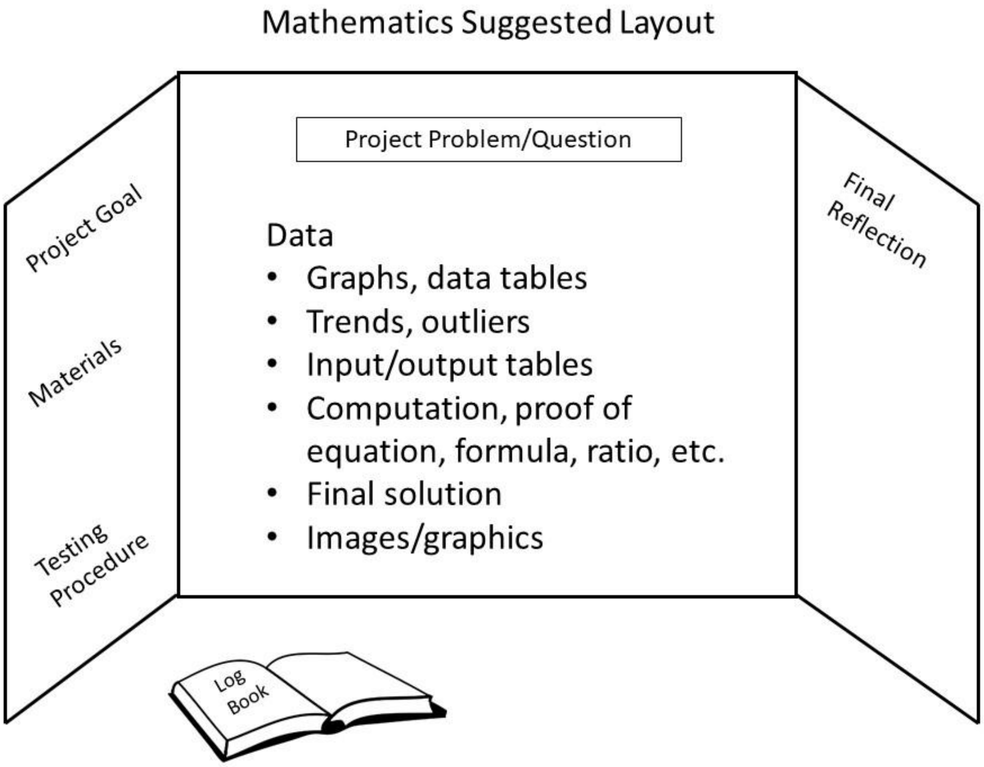
Mathematics

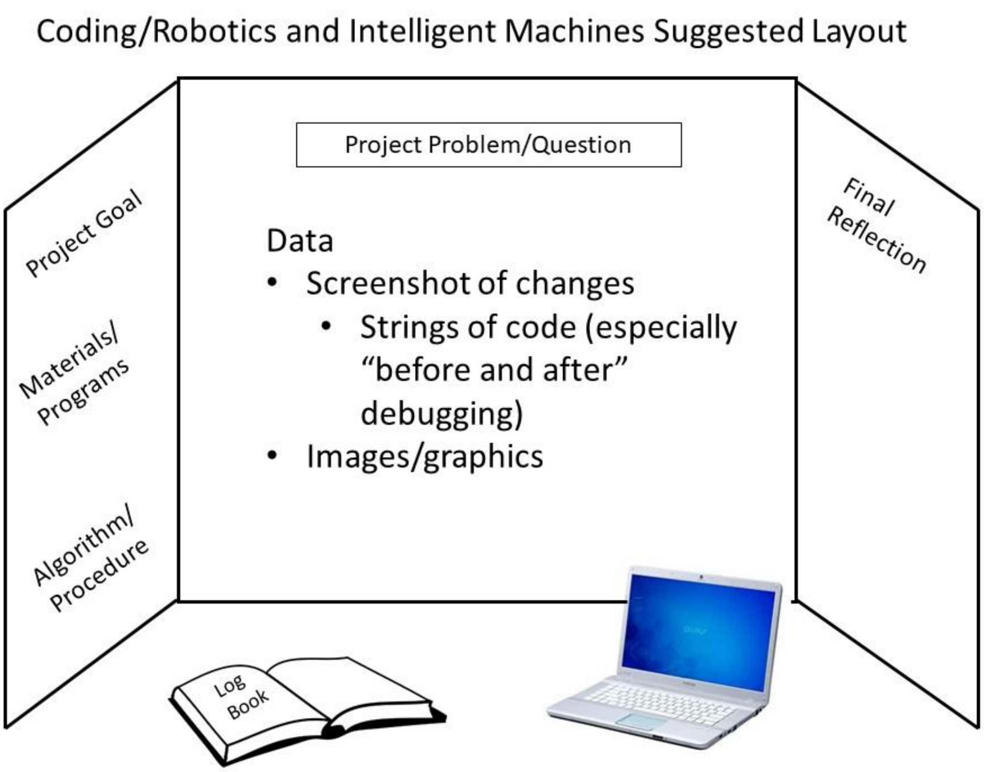
- process for developing formula, ratio, or solution
- testing procedure
- graphs of data
- trends
- explanation of outliers in your data
- final reflection

Coding/Robotics and Intelligent Machines

- coding language choice (Scratch, Arduino IDE, MakeCode, Tynker, JavaScript, HTML5, Xcode, C++, etc.)
- debugging process
- screenshots of code
- choice of specific important string of code
- modifications made
- final reflection
- **A printed copy of your entire final code or a computer that can access the code must be available during judging.**
- **You must bring your device to your interview to demonstrate how your program works. If you constructed or used a robot/intelligent machine, you must also bring that to your interview.**

Below are samples of a Mathematics and a Coding/Robotics and Intelligent Machines Project Display Board. Your board does not have to match this exactly, but it MUST have your problem and tell the story of your project.





SEF Student Checklist

Mathematics and Computer Science Division

Student Name____

check each box	Congratulations on completing your project! Use your SEF Student Handbook and this checklist to be sure you have completed all of the required parts and that you stayed within the rules for your project. ***In order to be safe and fair, if you don't follow the rules, your project will not be permitted in the Regional Science and Engineering Fair.***
☐	Check the box if you have completed and signed all of the ***necessary forms*** for your project. Look on page 2 of your Project Approval Form for what you need.
☐	Check the box if your ***project problem/question*** is on the board or in your log.
☐	Check the box if your ***project goal*** is on the board or in your log.
☐	Check the box if your ***materials and/or programs list*** is on the board or in your log.
☐	Check the box if your ***algorithm/procedure*** is on the board or in your log.
☐	Your board shows changes you made as you developed your program/equation. You can use screenshots as you ***debug and modify*** your program. You can add graphs and tables to show trends in data. Check the box if you have ***strings of code/graphs*** displayed on your board that show changes.
☐	Check the box if your ***final reflections*** are on the board.
☐	Check the box if your ***bibliography*** is complete, with at least ***three sources,*** and with your project.
☐	A ***log book*** is required for each student programmer (team projects require a log book for each student). Check the box if ***your log book is complete and with your project***.
☐	You can use ***photographs,*** (even ones that show your face), but you have to tell who took the photos. If the same person took them all, just put one label that says, "All photos taken by _______." But only use first names. You can also say, "Scientist's mom took this photo," or "Photo taken by scientist." Check the box if you have labeled your photographs.
☐	Items used from the ***Internet*** (articles, graphs, charts, pictures, etc.) need to have labels to cite the source. For example, "This chart was from (URL of website)." Check the box if you've labeled your Internet sources (if this applies to you.)
☐	Check the box if your display board is able to ***fold and lay flat*** and does not contain prohibited objects (such as lighting, soil, rocks, liquids, living or dead organisms, blood, sharp objects, plastic bottles, etc.).
☐	***Computer Science only:*** Check the box if you are prepared to bring a robot and/or computer to show the judges your code **OR** you have printed out your entire final code.
☐	Check the box if your project ***meets all of the rules and requirements*** outlined in the *SEF Student Handbook*.

Judging Criteria: *Mathematics and Computer Science Division*

______________________/____

Project Number *Category*

Project Problem____

	Superior	Very Good	Good	Poor	Notes
Research ● clear and focused ● description of practical need, problem to be solved, or purpose	10	8	4	2	
Design and Methodology ● well-designed plan for creating and testing program/equation ● explanation of choice of coding language/equation	15	10	5	2	
Testing and Modifying ● reflection throughout process *For Mathematics* ● the formula solves the problem ● trends are identified *For Computer Science* ● explanation of method of debugging program ● efficiency of code ● robot/device functions properly, if applicable	15	10	5	2	
Representation of Data/Design *For Mathematics* ● graphs identify trends ● clear graph of formula *For Computer Science* ● clarity of screenshots/graphics ● significance of coding strings represented	10	8	4	2	
Log Book ● dated entries/daily reflections ● sketches/diagrams/flow maps/world design ● access to entire code/formula/algorithm ● evidence of research ● bibliography (at least 3 sources)	15	10	5	2	
Interview ● clear, concise response to questions ● reflection on process/mathematical reasoning ● ideas for future research ● lessons learned ● If team, both members demonstrate significant contribution to project *For Computer Science* ● ability to connect specific code to task	15	10	5	2	
Display ● logical organization of project content ● tells story of project ● displays student learning ● device to demonstrate robot and/or program, if applicable	10	8	4	2	
Creativity ● project demonstrates imagination and inventiveness ● project opens up new possibilities or new alternatives	10	8	4	2	

**Form to be printed in yellow for the Regional Science and Engineering Fair.

Total ____________