

# Programming

---

[Introduction](#) | [Visual Programming](#) | [Scratch Programming](#) | [Programming Trends](#) | [Careers](#)

---

## Introduction to Programming

Computer programming (often shortened to programming) is a process that leads from an original formulation of a computing problem to executable programs. It involves activities such as analysis, understanding, and generically solving such problems resulting in an algorithm, verification of requirements of the algorithm including its correctness and its resource consumption, implementation of the algorithm in a target programming language, testing, debugging, and maintaining the source code, implementation of the build system and management of derived artifacts such as machine code of computer programs. The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solve a given problem.

According to a growing number of experts, learning computer science will not only pave the way for future employment prospects — job growth in this sector is booming — but help to accelerate the United States economic recovery. Learn about a new "superpower" that isn't being taught in 90% of US schools. The video stars Bill Gates, Mark Zuckerberg, will.i.am, Chris Bosh, Jack Dorsey, Tony Hsieh, Drew Houston, Gabe Newell, Ruchi Sanghvi, Elena Silenok, Vanessa Hurst, and Hadi Partovi.

Code is the foundation of computing. Whether you are using a social media app on your smartphone or working with a cloud server's API, the task relies heavily on a programming language. Ada Lovelace is recognized as the world's first programmer. She wrote an algorithm for Charles Babbage's Analytical Engine. Lovelace's contributions to computing marked the beginning of a rich history in programming despite the first programming languages predate the modern computer.

The first programming languages designed to communicate instructions to a computer were written in the 1950s but the first

actual modern programming language is hard to identify. There are four groups of programming languages and reflect the historical development of computer languages:

- machine languages
- assembly languages
- high-level languages
- fourth-generation languages

**Machine code** is the only language a microprocessor can process directly without a previous transformation. Currently, programmers almost never write programs directly in machine code, because it requires attention to numerous details which a high-level language would handle automatically, and also requires memorizing or looking up numerical codes for every instruction that is used.

**Assembly language** has no semantics and no specification, being only a mapping of human-readable symbols, including symbolic addresses, to opcodes, addresses, numeric constants, strings and so on. Typically, one machine instruction is represented as one line of assembly code. Assemblers produce object files which may be linked with other object files or loaded on their own.

**High-level** programming language is a programming language with strong abstraction from the details of the computer. In comparison to low-level programming languages, it may use natural language elements, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. memory management), making the process of developing a program simpler and more understandable relative to a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is

A **fourth-generation** programming language (4GL) is a computer programming language envisioned as a refinement of the style of languages classified as third-generation programming language (3GL). Each of the programming language generations aims to provide a higher level of abstraction of the internal computer hardware details, making the language more programmer-friendly, powerful and versatile. While the definition of 4GL has changed over time, it can be typified by operating more with large collections

of information at once rather than focus on just bits and bytes. Languages claimed to be 4GL may include support for database management, report generation, mathematical optimization, GUI development, or web development. Fourth-generation languages have often been compared to domain-specific languages (DSLs). Some researchers state that 4GLs are a subset of DSLs.

**Assignment:** Question for Thought 1

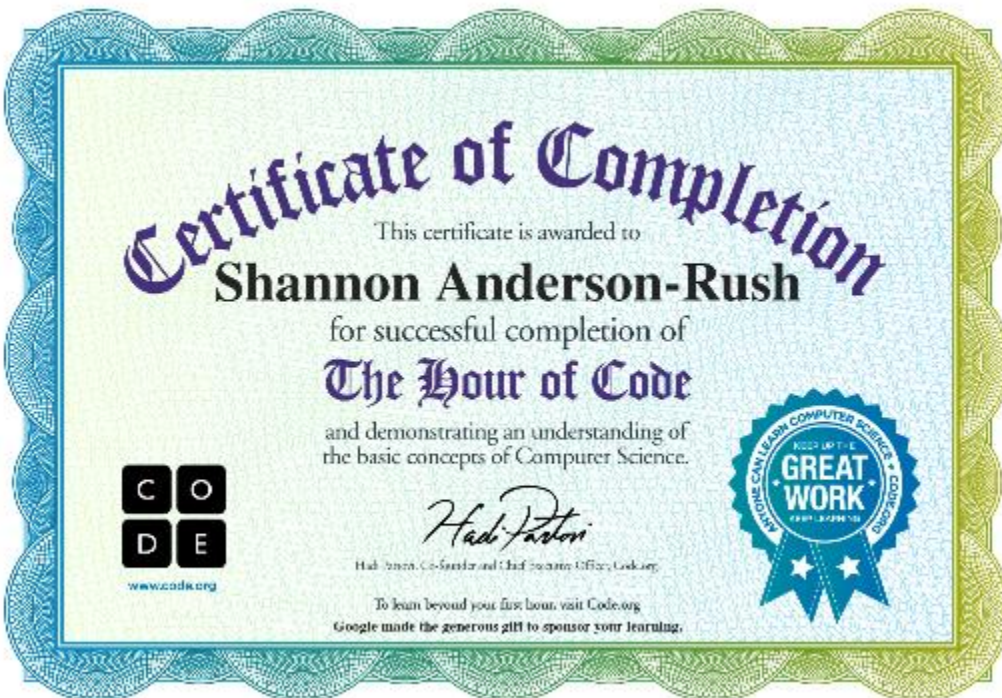
**Directions:** Create a list of 10 popular programming languages in use today and describe which industry or industries they are primarily used in and why. Place your list and description directly in the textbox below. Do not attach a separate document but be sure you proofread before selecting the submit button.

**Assignment:** Question for Thought 2

**Directions:** Describe three different programmed devices you rely on every day. Place your description directly in the textbox below. Do not attach a separate document but be sure you proofread before selecting the submit button.



## Visual Programming



As of March 27, 2015, a total of 108,402,167 people have participated in an hour of code and students had written 5,808,367,210 lines of code. That is an incredible number of people learning how to code, wouldn't you agree?

Launched in 2013, Code.org® is a non-profit dedicated to expanding participation in computer science by making it available in more schools, and increasing participation by women and underrepresented students of color. Our vision is that every student in every school should have the opportunity to learn computer science. We believe computer science and computer programming should be part of the core curriculum in education, alongside other science, technology, engineering, and mathematics (STEM) courses, such as biology, physics, chemistry and algebra.

### **Assignment: Code with Anna and Elsa**

**Code with Anna and Elsa**  
Code.org

Let's use code to join Anna and Elsa as they explore the magic and beauty of ice. You will create snowflakes and patterns as you ice-skate and make a winter wonderland that you can then share with your friends!

Ages 8+ | Modern browsers + tablets

9,770,943 participants

<http://hourofcode.com/frzn>

Teachers: Read important educator notes here

Go

**Directions:** Over nine million people have learned to code with Anna and Elsa. If you have not done so already, watch the video above about how to create a snowflake with Anna and Elsa. Let's use code to join Anna and Elsa as they explore the magic and beauty of ice.

Go to the [Anna and Elsa Hour of Code interactive website](http://hourofcode.com/frzn). You will use code to join Anna and Elsa as they explore the magic and beauty of ice. You will create snowflakes and patterns as you ice-skate and make a winter wonderland that you can then share with your friends! Follow the instructions to create a beautiful snowflake with Anna and Elsa. When you are finished, you will be awarded with an Hour of Code certificate like mine above. Take a screenshot of your certificate and submit the screenshot to itsLearning. Have fun.

## Assignment: Code Your Own Flappy Game

### Make your own apps or games

**Make a Flappy game**  
Code.org

Use drag-and-drop programming to make your own Flappy Bird game, and customize it to look different (Flappy Shark, Flappy Santa, whatever). Add the game to your phone in one click.

Ages 4-104 | Modern browsers, smartphones, tablets

7,517,078 participants

<http://hourofcode.com/flap>

Go

**Directions:** Over seven million people have learned to code their own Flappy game. If you have not done so already, watch the video above about the Flappy game.

Go to the [Flappy game Hour of Code interactive website](#). Follow the instructions to create a fun flappy bird type game that you will be able to upload to your phone or other device. When you are finished, you will be awarded with an Hour of Code certificate like mine above. Take a screenshot of your certificate and submit the screenshot to itsLearning. Also include a reflection (about 50 words) of what you learned. Have fun.

---

## Scratch Programming

Visual programming is the creation of a computer program by utilizing pictorial elements. Traditionally, a program is a sequence of text statements used to achieve a certain result or solve some problem. Programming languages often have particular ways of representing the work to be done thus leading to complexity. Visual programming attempts to make the creation of programs simpler.

The ability to code computer programs is an important part of literacy in today's society. When people learn to code in Scratch, they learn important strategies for solving problems, designing projects, and communicating ideas. Coding isn't just for computer whizzes, says Mitch Resnick of MIT Media Lab - it's for everyone. In a fun, demo-filled talk Resnick outlines the benefits of teaching kids to code, so they can do more than just "read" new technologies - but also create them.



Click on the picture to go to the Scratch website. The video is in the upper right corner.

## Assignment: Scratch - Try It Out

**Directions:** Go to the [Step-by-Step Intro](#). Follow the instructions on the right to create your first Scratch project. Follow the [instructions](#) on how to turn in a Scratch project.

## Assignment: Scratch Cards

**Directions:** Complete the Scratch cards located in itsLearning. You can ignore the directions regarding printing and folding the cards. Follow the short instructions on working with the different elements in Scratch. You will **not** submit the Scratch cards for grading.

## Assignment: Scratch Greeting Card

**Directions:** Create a Scratch project that reflects an animated greeting card. The greeting card could be for a holiday, birthday, get well, etc., and must include at least two sprites and a backdrop. One of the sprites should change colors when the space bar is pressed and one of the sprites should dance to a beat. The dancing sprite should have at least three costumes. The sprite must convey a message that reflects a greeting based on the purpose of the card.

Follow the [instructions](#) on how to turn in a Scratch project in the itsLearning assignment. You should also download the full assignment with [rubric](#).

## Assignment: Scratch Game

**Directions:** You are going to create a video game using Scratch! Download the Create Your Own Game in Scratch project from itsLearning. Follow the project instructions. Follow the [instructions](#) on how to turn in a Scratch project. When you submit your project link, you must also answer the questions listed with the project.

---

## Programming Trends

Programming trends are driven by greater efficiency, increased customization, and ease-of-use. The new technologies that deliver one or more of these trends eclipse the previous generation. Programming is an endlessly fascinating profession: rapid change, passionate debate, sudden comebacks. According to Peter Wayner at *InfoWorld*, the 15 hot programming trends are:

- Preprocessors: It wasn't long ago that people who created a new programming language had to build everything that turned code into the bits fed to the silicon. Then someone figured out they could piggyback on the work that came before. Now people with a clever idea just write a preprocessor that translates the new code into something old with a rich set of libraries and APIs.



### What's trending in programming?

- JavaScript MV frameworks: Long ago, everyone learned to write JavaScript to pop up an alert box or check to see that the email address in the form actually contained an @ sign. Now HTML AJAX apps are so sophisticated that few people start from scratch. It's simpler to adopt an elaborate framework and write a bit of glue code to implement your business logic. There are now dozens of frameworks like Kendo, Sencha, jQuery Mobile, AngularJS, Ember, Backbone, Meteor JS, and many more -- all ready to handle the events and content for your Web apps and pages.



- CSS frameworks: Once upon a time, adding a bit of pizzazz to a Web page meant opening the CSS file and including a new command like *font-style:italic*. Then you saved the file and went to lunch after a hard morning's work. Now Web pages are so sophisticated that it's impossible to fill a file with such simple commands. One tweak to a color and everything goes out of whack. It's like they say about conspiracies and ecologies: Everything is connected. That's where CSS frameworks like SASS and its cousins Compass have found solid footing. They encourage literate, stable coding by offering programming constructs such as real variables, nesting blocks, and mix-ins. It may not sound like much newness in the programming layer, but it's a big leap forward for the design layer.
- SVG + JavaScript on Canvas: Now that the JavaScript layer has the ability to do much of the same, browser manufacturers and developers are cheering for the end of Flash. They see better integration with the DOM layer coming from new formats like SVG (Scalable Vector Graphics). The SVG and HTML comprise one big pile of tags, and that's often easier for Web developers to use. Then there are large APIs that offer elaborate drawing on the Canvas object, often with the help of video cards. Put them together and there few reasons to use Flash anymore.
- Almost big data instead of big data
- Game frameworks: Most games developers today use libraries like Unity, Corona, or LibGDX to build their systems. They don't write C code as much as instructions for the libraries. Is it a shame that our games aren't handcrafted with pride but stamped out using the same engine? Most of the developers are relieved -- because they don't have to deal with the details, they can concentrate on the game play, narrative arc, characters, and art.

- Single-page Web apps: Instead of websites with multiple pages, Web apps are popular. New Web apps are front ends to large databases filled with content. When the Web app wants information, it pulls it from the database and pours it into the local mold. There's no need to mark up the data with all the Web extras needed to build a Web page. The data layer is completely separate from the presentation and formatting layer. Here, the rise of mobile computing is another factor: a single, responsive-designed Web page that work like an app -- all the better to avoid the turmoil of the app stores.

Read Peter Wayner's full article on [15 hot programming trends -- and 15 going cold](#). What do you think has prompted the current trends in programming? (Hint: Think about what we use to access programs.)

---

## Careers

A computer programmer creates and writes the code for software applications and operating systems. After a software developer or computer software engineer designs a computer program, the programmer writes code that converts that design into a set of instructions a computer can follow. The programmer tests the program to look for errors and then rewrites it until it is debugged, or error-free. A programmer continues to evaluate programs that are in use, making updates and adjustments as needed.

### Assignment: Question for Thought 3

**Directions:** Find out about three career opportunities in programming. Pick one and find out the education, training, and experience required. In a paragraph (about 150 words) list the three career opportunities and then expand upon of the three. Explain why this career might be of interest to you. Place your paragraph directly in the textbox below. Do not attach a separate document but be sure you proofread before selecting the submit button.

---

## Resources

If you are having problems viewing this page, opening videos, or accessing the URLs, the direct links are posted below. All assignments are submitted in itsLearning. If you have having problems, contact Mrs. Rush through the itsLearning email client.

Hour of Coding video: [http://www.youtube.com/watch?v=r990CWYtx\\_o](http://www.youtube.com/watch?v=r990CWYtx_o)

Code Stars: <http://www.youtube.com/watch?v=dU1xS07N-FAf>

Mitch Resnick: Let's Teach Kids to Code:

[http://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code?utm\\_content=awesm-publisher&utm\\_campaign=&utm\\_source=direct-on.ted.com&awesm=on.ted.com\\_MResnick&utm\\_medium=on.ted.com-static](http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?utm_content=awesm-publisher&utm_campaign=&utm_source=direct-on.ted.com&awesm=on.ted.com_MResnick&utm_medium=on.ted.com-static)

Scratch Introduction video: <http://scratch.mit.edu/#>

15 Hot Programming Trends article: <http://www.infoworld.com/print/233343>

Anna and Elsa Snowflake: <http://studio.code.org/s/frozen/>

Flappy Game Hour of Code: <http://studio.code.org/flappy/1>

Review: <https://www.goconqr.com/en-US/p/2388108-Introduction-to-Program-Review-quizzes>

Transcript: <http://mrsrush.net/programming/index.pdf>

---

[Transcript of this lesson](#)