

Learn Computational Thinking using Scratch: Level 1

Important: Read "Scratch-courses-learning-objectives.pdf" before reading this document.

Summary:

This document describes the high level curriculum of the course. It is presented in the form of how the day-to-day flow might look like while running this course.

Duration:

The level 1 course can run anywhere from about 25 hours (which is the minimum to make it meaningful) all the way to 50 hours if you use the rich content available and include a "capstone" project at the end. As a teacher/learner, you need to pace and select the content according to your calendar. How the curriculum gets divided among sessions would depend on:

- Length of each session (which should be between 60 to 90 minutes)
- Level of students (grasp, speed, interest)

References:

This document refers to different external resources as listed below:

- The textbook "**Learn CS Concepts with Scratch**". This is the most important resource necessary for this course since the curriculum relies primarily on this book. References to chapters, review questions and programming assignments made in the curriculum are from this text-book unless otherwise specified.
- **PowerPoint** slides and **demo program files** are provided under the "Concepts" folder.
- **Sample projects** are provided under the "Projects" folder.
- The book "**Practice CS Concepts with Scratch**" provides numerous additional review questions and programming assignments some of which are mentioned in the curriculum below. This book is an important resource to improve the effectiveness of this course.
- The book "**Pen Art in Scratch Programming**" is an excellent resource to go deeper into Turtle Geometry and creative drawing using the Pen feature of Scratch.

Curriculum

Unit 1:

Learn new concepts:

- Scratch UI: (Concepts\Scratch UI\Scratch UI.pptx)
 - o layout, create/save projects
 - o Sprites

- Stage and backdrops
 - Command blocks and Scripts
 - Cursor tools
- Paint editor (Concepts\PaintEditor\PaintEditor.pptx)
 - Demo of fill effects: fill-effects.sb2, objects.sb2

Programming practice:

- Programming practice Chapter 1.1

Learn new concepts:

- Sequence (Concepts\sequence\sequence.pptx and demo programs)
- Audio features in Scratch:
 - Playing sound files: Concepts\Audio\Audio-features-1.pptx and demo.sb2
 - Creating music: Concepts\Audio\Audio-features-2.pptx

Programming practice:

- Programming practice Chapter 1.2

Learn new concepts:

- Green flag: running two sprites at the same time
 - how to build a simple dialog program
 - Timeline-based synchronization

Programming practice:

- Knock-knock joke using Timeline-based synchronization (demo programs in Concepts\Synchronization-using-timeline)
 - Solution: Knock-knock-1.sb2 (Book: "Practice CS Concepts")
- Knock-knock joke (add audio)

Learn new concepts:

- Simple looping (Concepts\Looping\Simple-looping.pptx, why-looping.sb2)
 - If on edge bounce, rotation styles
 - How to build a simple motion program (to and fro and spinning)
- Smooth motion using repeat

Learn new concepts:

- Animation using costumes (Concepts\costumes\animation-with-costumes.pptx, Creating Costumes.pptx, demo programs)
 - Multiple backdrops

Programming practice:

- Programming practice Chapter 1.3

Learn new concepts:

- XY geometry (Concepts\XYGeometry\XYGeometry.pptx)
- Nested looping (multiple jumps) (Concepts\Looping\Nested-looping.pptx)

Programming practice:

- Different ways to jump up and down
- Programming practice Chapter 1.4

Mini-project: About-me or Monologue

- Specification – Chapter 1.5
- Examples:
 - o Aboutme-1.sb2
 - o Aboutme-2.sb2

Learn new concepts:

- Relative/absolute motion (Concepts\Motion\Motion.pptx)
- Smooth motion using repeat
- 3-D effect using change size
- Synchronization using broadcasting (Concepts\Broadcasting\Broadcast.pptx)
 - o Example of a dialog

Programming practice:

- Do the same dialog (as above) using "broadcast" instead of "broadcast and wait"
- Programming practice Chapter 1.6

Review quiz:

- Unit 1 review questions

Programming project:

- Project: Nursery rhyme or short story
 - o Demo: dragon-bus-1.7.sb2 (solutions.zip)
- Show specification (Concepts\Short-animation\short-story-spec.docx)
- Explain the idea of design checkpoints (design-checkpoints.docx)
- Use rubric for evaluation (rubric.xlsx)

Unit 2

Learn new concepts:

- Concurrency (Concepts\Concurrency\Concurrency.pptx)

Programming practice:

- Programming practice Chapter 2.1: 1 and 2 (many things in parallel, rolling ball)

Learn new concepts:

- Reset script: to set original states of sprites
 - o Demo: concepts\reset\demo.sb
- Events (Concepts\Events\Events.pptx)
- User interaction using keyboard (Concepts\User interaction\User interaction-basic.pptx)

Programming practice:

- (Chapter 2.3) Helicopter program (with costumes/clouds, without helipad/gravity)
- Helicopter program (add gravity)

Learn new concepts:

- Graphical effects (Concepts\Graphical Effects\Graphical Effects.pptx, demo.sb2)
- Intro to conditional statements (Concepts\Decisions\1-Conditionals-IF-ELSE.pptx)
 - o Conditions
 - o Sensing touch
 - o IF and IF-ELSE

Programming practice:

- Programming practice Chapter 2.2: 1 (Simple pong program, no stopping)
- Helicopter program (with helipad)

Common questions/points:

- How to move the ball up-down (use direction)
- "If touching" doesn't work (it must be inside forever)
- Use "touching sprite" in place of "touching color" wherever possible.
- Don't use "forever" unnecessarily.

Learn new concepts:

- How to follow the pointer (Concepts\User interaction\User interaction-basic.pptx)

Programming practice:

- Programming practice Chapter 2.2: 2 (color changing animal)

Program review questions:

- Helicopter: clouds, gravity, landing
- Conditionals (touching mouse pointer/edge etc.)
- Explain helicopter landing

Learn new concepts:

- STOP commands (Concepts\Decisions\3-Conditionals-STOP.pptx)

Programming practice:

- Programming practice Chapter 2.2: 1 (Simple pong program, with stopping when touching the ground)

Review quiz:

- Unit 2 review questions

Unit 3

Re-cap concepts:

- User interaction using mouse pointer

- Following the pointer

Programming practice:

- Programming practice Chapter 3.1: 1 (jump on mouse), 2 (run away)

Learn new concepts:

- Click-buttons using broadcasting

Programming practice:

- Programming practice Chapter 3.1: 3 (Boy and Wizard)

Learn new concepts:

- Variables (Concepts\Variables\1-Variables-intro.pptx)

Programming practice:

- Count 1, 2, 3 ... infinity!
- Write a simple never-ending animation (anything at all) and use a variable to implement a timer of say 30 seconds. The animation should stop after the time is over.

Learn new concepts:

- Multiple Ifs (Concepts\Decisions\IF-ELSE-combinations.pptx)
 - o One after the other: (combine the IFs in Pong)
 - o Nested: equivalent to AND

Learn new concepts:

- Wait until (Concepts\Decisions\2-Conditionals-WAIT.pptx)

Programming project:

- Maze game (Chapter 3.3)
- Basic features:
 - o A simple maze
 - o Clearly visible "Start" and "Finish" locations
 - o Prizes to eat along the way
 - o Pacman moves with arrow keys
 - o Touching the walls of the maze makes pacman go back to Start

Programming project:

- Maze: Advanced features:
 - o Moving (or spinning) obstacles, when touched pacman goes back to Start
 - o A monster follows pacman forever; when touched pacman goes back to Start
 - o Use a variable to track prize points and penalties.
 - o Timer

Learn new concepts:

- How to include "multiple" levels of challenge/difficulty in a game (Concepts\Variables\Multilevel-games-using-variables.pptx.pptx)

- Demo: maze-multilevel.sb2

Pen Programming:

At this stage cover the Pen commands (see Unit 7 of the textbook) and use "Pen Art in Scratch Programming" as additional reference. In addition, there are lots of programming problems provided in Unit 7 of "Practice CS Concepts with Scratch".

Learn new concepts:

- Pen commands (Concepts\Pen art\Pen art.pptx)
 - Color-codes.sb2 shows possible color-shade combinations
 - Demo-various-designs.sb2 shows different designs
- Explain difference between "set" and "change"
- Sprite can hide while drawing

Programming practice:

- Pen Art Book: Chapter 1: problem #2 (collage of various shapes)
- Additional practice: Etch-a-sketch-1 (from "Practice CS Concepts")

Learn new concepts:

- Applying power of programming to Pen Art:
 - Looping

Programming practice:

- Programming practice Chapter 7.1: 1 (repeat patterns)

Program review:

- Repeat patterns:
 - Repeat patterns include "hidden" things like turns and gaps
 - Test before using "repeat"
 - Do not use "absolute" motion
 - Basic pattern involves:
 - Drawing basic shape
 - Getting ready to draw it again

Learn new concepts:

- Turtle round trip (Concepts\Pen art\trt-demo.sb2)

Programming practice:

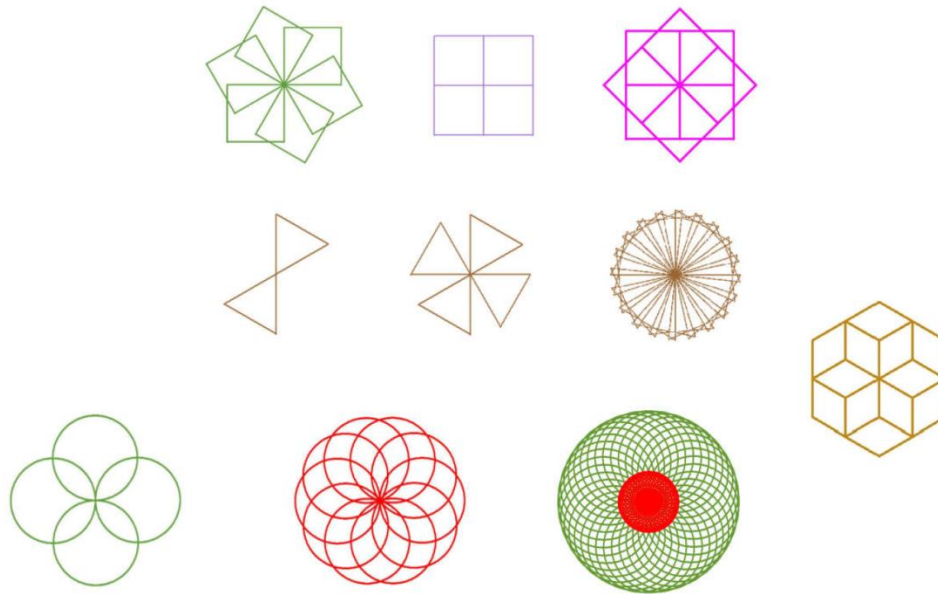
- Programming problem "shapes-2" in Unit 7 of "Practice CS Concepts with Scratch"
- Draw a circle.

Learn new concepts:

- Nested looping (Concepts\Looping\Nested-looping.pptx)

Programming practice:

- Draw the flower designs shown below: (Hint: They all use basic shapes such as square, triangle, circle, etc.)



- Additional practice: Wall of bricks (from "Practice CS Concepts")

Review quiz:

- Unit 3 review questions

Capstone project:

Let students design and implement a Scratch program of their choice. Ask them to form small teams (min 2, max 4) and submit a project proposal. The proposal should include how they plan to divide the work among team members. Please ensure the project is sufficiently challenging and requires the application of all the important CS concepts and Scratch features learnt in this course.

Author: Abhay B. Joshi (abjosshi@yahoo.com)

Last updated: 22 May 2019

Courses created and published by: SPARK Institute and Publications

Copyright notice: There is no copyright on this document or the files referred in it; this material is freely available to anyone who is interested in learning or teaching Scratch programming.

