

Course A: Foundations of Computer Science

# Overall Course Goals

This course is designed for students who can read and have no prior programming experience. In this course students will create programs to solve problems and develop interactive games or stories they can share.

# CS Themes

Programs, algorithms, loops, debugging, conditionals, binary, events

# Lesson Sequence

Instructors will teach seven unplugged activities from the [www.code.org Elementary Course 2](http://studio.code.org/s/course2). Instructors will also include computer puzzles as in class activities or homework assignments. Use the PowerPoint for each unit as a guide through each lesson.

## Week One

Ask girls what they already know about computer programming and what they hope to learn from this course. Pass out materials, such as folders and bags, to each student. See [Administrative](#_Administrative_Notes) section for further information.

In order to understand programming, students will color squares on graph paper in an effort to reproduce an existing picture.

* Stage 1: Graph Paper Programming
* Coding Concepts: algorithm and program
* Materials Needed: pencils
* Homework: Stage 3: Maze Sequence

## Week Two

A guest speaker should present for about 20 minutes on computer programming.

Students will relate the concept of algorithms back to everyday real-life activities by making paper airplanes.

* Stage 2: Real-Life Algorithms: Paper Planes
* Coding Concepts: algorithms and sequencing
* Materials Needed: pencils, paper, scissors, glue sticks
* Homework: Stage 4: Artist Sequence

## Week Three

Students will practice converting sets of actions, such as dance moves, into a single loop.

* Stage 5: Getting Loopy and Stage 6: Maze Loops
* Coding Concepts: loops
* Materials Needed: pencils, paper, scissors, glue sticks
* Homework: Stage 7: Artist Loops and Stage 8: Bee Loops

## Week Four

A guest speaker should present for about 20 minutes on computer programming.

Students will complete a short review of Graph Paper Programming, then, in teams, will race to create a program, one instruction at a time.

* Stage 9: Relay Programming and Stage 10: Bee Debugging
* Coding Concepts: programming and debugging
* Materials Needed: pencil, paper
* Homework: Stage 11: Artist Debugging

## Week Five

Students will demonstrate how conditionals can be used to tailor a program to specific information. Students will use conditionals as they play a game using cards.

* Stage 12: Conditionals
* Coding Concepts: conditionals
* Materials Needed: pencil, paper, at least 5 decks of playing cards
* Homework: Stage 13: Bee Conditionals

## Week Six

A guest speaker should present for about 20 minutes on computer programming.

Students will learn how to take something that we know and translate it into a series of ons and offs in order to understand binary.

* Stage 14: Binary Bracelets
* Coding Concepts: binary
* Materials Needed: pencil, colored pencils, scissors, tape.
* Homework: Binary Game <http://britton.disted.camosun.bc.ca/binary.swf> (Show students how to play the game before they go home and send out an email with the link.)

## Week Seven

Students will learn how to add variety to a pre-written algorithm using events.

* Stage 15: The Big Event and Stage 16: Flappy
* Coding Concepts: events, sequencing, loops
* Materials Needed: pencil
* Homework: Make-up any missed homework assignments

## Alternatives

Students will go to a high school to program Lego Mindstorm NXT Robots to complete various tasks.

* Coding Concepts: programming, loops, sequencing
* Materials Needed: signed permission slip
* Homework: Make-up any missed homework assignments

## Week Eight

Students will have the opportunity to apply all of the coding skills they've learned to create an animated story.

* Stage 17: Play Lab and Stage 19: Artist Nested Loops
* Coding Concepts: programming, events, sequencing, loops, conditionals
* Materials Needed: pencil
* Homework: None
* Extension: Teach Stage 18: Your Digital Footprint or introduce the students to Scratch

# Online Resources

 [www.code.org](http://www.code.org)

# Visitors/Projects/Extensions

Guest speakers recommended for this course would be programmers who create animations or interactive tools, especially for entertainment. Extension for in class learning are listed in each lesson plan on the code.org website. At home, students should extend their learning by competing the puzzles as homework. Students can also extend their learning by creating their own projects in these two code.org websites: <http://studio.code.org/p/playlab> and <http://studio.code.org/p/artist>.

# Administrative Notes

Spend time getting to know the code.org website and review the curriculum.

Before the first class, the class should be created and student names should be entered on code.org. Do this by clicking “Students Accounts and Progress” and then clicking “New Section.” The log in type should be picture and the course should be course 2. Click “Manage Students” to add new students to the class. At the bottom of the page, click “Print out cards with your students’ login information” to print student log-in cards and tape them to the inside of their folders.

Assign computers to the students and write their computer numbers on their folders. On the first day, have the students bookmark the link to code.org section’s sign-in page.

Monitor students’ progress by clicking “Students Accounts and Progress” and then clicking “View Progress.”

Every class, students should work on the computers for at least 20 minutes to start the homework. During weeks three, four, and eight, students should have additional time on the computers to complete puzzles during class.