

Midterm 2 Review

CS 107, spring '21

The second midterm will focus **primarily on material since the first midterm**, but it is comprehensive and will cover concepts and code from the entire semester.

The format for Midterm 2 will be similar to that of Midterm 1. It will consist of:

- 1) **short-answer conceptual** questions,
- 2) questions asking you to **trace code blocks** to determine how components and variables will change, and
- 3) **coding problems** which are similar to labs done in class but different.

The test is **closed book and notes, closed internet and communication**. You will be able to open Thunkable and your own apps within it, but that and your test are the only thing you should view or interact with.

How to Study:

- (Re-) read the chapters
- Review your coding labs and projects. You can open these up during the test and tweak them. Test yourself to see if you can code them from scratch so that you know you understand them.
- Go over lessons and worksheets, practice writing out answers to conceptual questions.
- Practice tracing code, jotting down how properties and variables change on each operation.
- Review quizzes and midterm 1.
- You are strongly encouraged to study with a group!

The questions will involve the topics listed below. The coding problems will be similar to the ones you've done in labs/projects, but modified so make sure you really understand how they work and how to apply them to slightly different problems (you can't just memorize)

Topics Since Midterm 1

Persistence and Cloud

Spreadsheet coding

each row has an auto-generated id which is not the same as an index
see [spreadsheet coding worksheet](#) and solution at bottom

Mapping the Data Viewer List to a Data Source

"get value" and "create new row" block

Fixed data vs. Dynamic Data

Persistent data [conceptual lesson](#)

how does an app behave differently if its data is persistent?

Midterm 2 Review

CS 107, spring '21

What is the significance of MediaDB and Cloudinary? What type of app behavior and persistence does it facilitate?

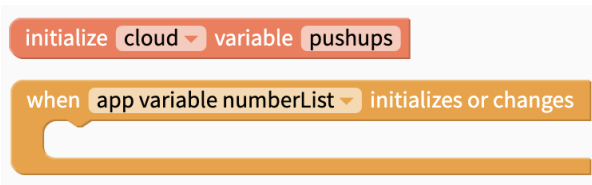
Stored and Cloud Variables

how do stored/cloud variables provide an alternative to spreadsheets for persistence?

how does an app's behavior change if you switch a variable from "app" to "stored"? or "cloud"?

You must set initial values for stored/cloud variables in the Screen.Opens (asking if value is null)

Explain the significance of the event handler below. How does an app behave differently when you put code in that event handler?



Pushups sample, QA chat sample

Chapter 9 is helpful for these concepts.

Repeat Loops

[repeat loops worksheet, solutions](#)

Mechanics of *count with*, *for each item in list*, and *repeat while*.

what operations are performed automatically for each kind of repeat block?

Sample repeat code:

- Add up numbers $1+2+\dots+n$, add up numbers in a list,
- factorial: $1*2*3*\dots*n$
- Count occurrences of something in a list, find minimum/maximum
- Build a text string from a list, e.g., build an equation from a list of numbers, or put "--" in between animals (see question 6 solution in [repeat loops worksheet](#))
- Be able to do repetitive operations like the ones listed here using the different type of repeat blocks

Chapter 8 is helpful for these topics

Midterm 2 Review

CS 107, spring '21

Functions

Be able to code with functions including those with input parameters.

How does functional abstraction help in problem solving and software engineering?

What are the benefits of organizing an app with functions?

What are the benefits of defining a parameter for a function?

What is meant by refactoring?

What is meant by “code reusable functions”? How do parameters help? Why is it important?

What are the mechanics involved with a function call? Given some code with functions (and calls), specify the order with which operations will be performed

Chapter 11 is helpful for these topics

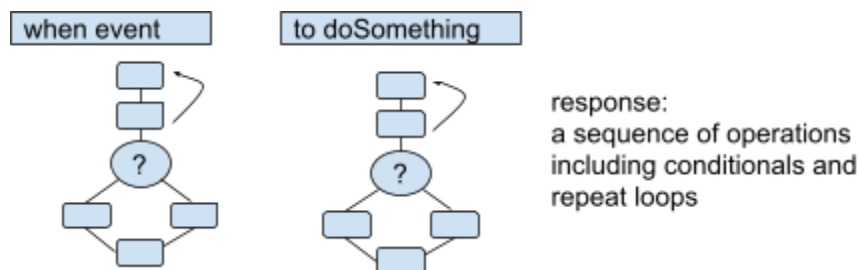
Tracing an App

This has been a focus and is a key to coding and debugging. Practice mechanically stepping through blocks, showing how each variable and property changes. You'll be asked to do this on the test!

Older Topics with some updates

App Grammar: What does an app's behavior consist of?

Chapter 1,2,3



- App behavior consists of a set of event handlers
- An event handler has an event and a response
- Response consists of a sequence of operations
- Operation is either a set, a call to a built-in function, **or a call to a programmer-defined function.**

Midterm 2 Review

CS 107, spring '21

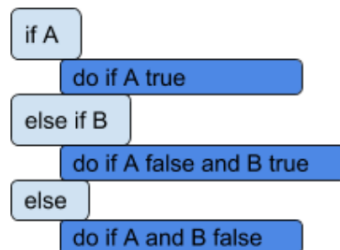
- Some operations are conditional (within an if) and **some are repeated**.
- Functions can have **input parameters** and a return value. Placeholder input parameters defined within the function are called **formal parameters**. The data sent into the function from the caller (put into a slot) are called **actual parameters**.

An app has hidden memory.

- What does the hidden memory consist of?
- Explain difference between the properties that appear in Designer and in hidden memory
- Does the UI change immediately when a property changes?
- Component properties and variables are named memory cells
- Properties are part of a component, variables are free agent memory cells
- Properties referred to as “component.property”
- What does “set” mean? What does “get” mean?
- How do you increment a value?
- When do you need to define a variable?
- Chapter 3 is especially helpful for these questions

Conditionals

- How do you ask questions in an app?
- How would you code something with 4 possible branches?
- if-elseif-elseif-else



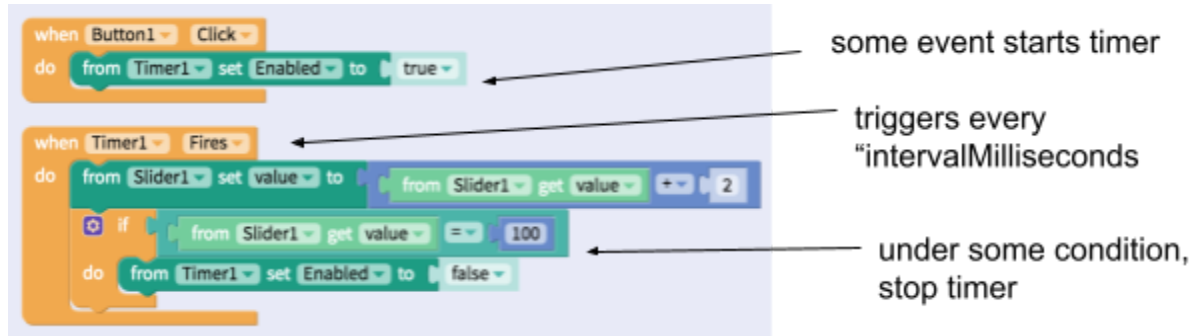
- boolean values
- and/or/not
- Chapter 3

Coding Timed (animated) Behavior

- Timer/Animation pattern
- When Timer Fires event handler
- Timer.Enabled
- Timer.Interval:
- example: make a slider move from 0 to 100

Midterm 2 Review

CS 107, spring '21



- Suppose the slider value starts at 0. How long will it take for the value to get to 100?

Tracing an app

Practice looking at code and stepping through it row by row. Using paper and pencil (or a spreadsheet) and show how each property and variable changes as you step through the event handlers.

Lists and Indexing

- List variable-- named set of memory cells
- index-- a variable used to keep track of the current position in a list, e.g., current quiz question
- incrementing the index, checking index vs. length of list
- get block to get an item from a list
- why important to use abstract "length of list" instead of a number
- ListView-- how to display a list
- Review various slideshow samples (wrap-around, disable next/prev)
- Synchronized lists and lists of lists
- Chapter 4

Calling Functions

- input parameters
- output parameters, scope of output parameters-- where can they be used
- nested calls
- use of Translator, ImageRecognizer, SpeechRecognizer, Camera
- Chapter 5

Text Operations

- join, get, length, upcase, lcase, Chapter 6

Canvas, Sprites, and Animation

- Key properties of sprites: x,y, speed, angle

Midterm 2 Review

CS 107, spring '21

- transporting and moving smoothly
- Key events of sprites (colliding with edges, other sprites)
- Chapter 7