



Digital Journal by Franklin Liu

Computer Science Essentials

Instructor: Matta,Jasneet Kaur

Unit 1

Creative Computing: Building with Blocks

Lesson 1.1: Introduction to Computer Science Essentials

Table of Contents

<u>Activity 1.1.1: Getting Started with Block-based Programming: Digital Doodle</u>	2
<u>Activity 1.1.2: Algorithms and Coding Fundamentals: Happy Accelerometer</u>	6
<u>Activity 1.1.3: Conditionals and Event-driven Programming: Happy Balance</u>	11
<u>Activity 1.1.4: Local and Global Variables: Guessing Game - Two Player</u>	14
<u>Activity 1.1.5: Iteration and Loops: Guessing Game - One Player</u>	19
<u>Project 1.1.6: App Development: Creative Expression</u>	22
<u>Activity 1.2.1 Problem Solving: Interview Database</u>	24
<u>Activity 1.2.3 Procedural Abstraction: Price Per Slice</u>	28
<u>Activity 1.2.4 Lists: Survey Says</u>	33

Activity 1.1.1: Getting Started with Block-based Programming: Digital Doodle

Essential Questions:

1. Why is it important to become a creator and not just a user?

It is important to become a creator and not just a user because it allows me to understand my computer on a deeper level and allows me to create apps when existing ones do not fully fit my needs.

2. How does block-based programming make life easier when coding?

Block-based programming makes life easier because it allows a programmer to visualize their code. They can easily see how the logic of the program flows.

3. Why are independent and cooperative strategies so important in computer science?

Independent and cooperative programming are both important because programmers often have to work in groups to do things faster. At the same time, they also often have to work alone on a project.

New vocabulary:

Code
Inputs
Outputs
Block-based Programming

New concepts:

- Programming Language Abstraction
- User-centered design
- Iterative Design and Debugging
- Event-driven programming

What are the feature requirements for your application?

- Push a button to take a picture
- Draw on the picture
- Change the color they draw with
- Change the line width

Signature: Franklin Liu	Date: 9/13/21	Cont. on Pg. 2-5
Witness: Ananya Mahadi	Date: 9/16/21	Proprietary Information

Debugging notes:

- Check for hardware or emulator issues
- Check for compilation errors
- Review comments and algorithms
- Make your code easy to read. Collapse unrelated code
- Debugging strategies:
 - Do It command
 - Disable command
 - Code trace, variable trace
- Test the app
- Fix the bug
- Test again

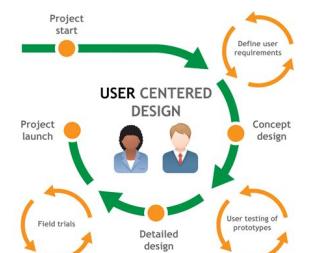
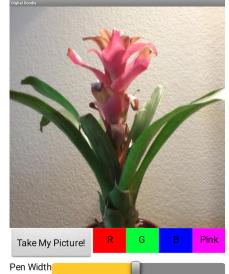
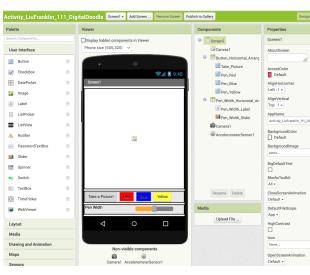
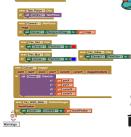
Screenshots:

Take a screenshot of the design view and block view and paste it here.



Signature: Franklin Liu	Date: 9/13/21	Cont. on Pg. 2-5
Witness: Ananya Mahadi	Date: 9/16/21	Proprietary Information

Notebook responses:

Term	Example	Meaning	Picture
User Story	A comment box on YouTube	A feature of an app that is visible as part of a user interface	
User Centered	A user-friendly interface	Optimized for the needs of the target audience	
User Interface	The look and feel of a mobile app	what a user interacts with on the device. Includes touchscreen, buttons & sensors.	
Designer view	MIT App inventor's designer view, where you drag components out of boxes and customize them to your preference	The view where you design the user interface	
Block view	Scratch's view, where you drag and drop blocks to make programs	The view where you program the program features	

Signature: Franklin Liu	Date: 9/13/21	Cont. on Pg. 2-5
Witness: Ananya Mahadi	Date: 9/16/21	Proprietary Information

Debugging	The GNU Debugger (gdb), which provides a terminal interface to debugging	The process of finding and fixing errors ("bugs")	
Iteration	A beta release of a program	A new version of a program	

Conclusion Questions:

Why is it important to design incrementally? Consider: During what iteration did you have a working app? What did you stand to lose and gain with each iteration?

It is important to design incrementally because it results in fewer bugs. When someone programs incrementally, they can fix the bugs in each increment reasonably easily. If they write the code all at once, then it becomes harder to track what does what and harder to debug.

Why do you think computer science professionals included things like communication, collaborations, and fostering an inclusive computing environment in the computer science practices?

Computer science professionals emphasize teamwork in the computer science practices because it is a very important part of software engineering. People often need to work together to build software, in order to make it a higher quality and to do it faster. Also, it is good to have multiple people looking at code because it results in more bugs being caught and fixed.

How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I interpreted and responded to the Essential Questions from the perspective of a programmer. All of my answers were from the perspective of how the questions would benefit the programmer.

Signature: Franklin Liu	Date: 9/13/21	Cont. on Pg. 2-5
Witness: Ananya Mahadi	Date: 9/16/21	Proprietary Information

Activity 1.1.2: Algorithms and Coding Fundamentals: Happy Accelerometer

Essential Questions:

- How do I describe algorithms to someone new to computer science?

I would describe algorithms by simplifying its steps and breaking it down into the parts that are repeated.

- What mathematical and logical concepts have I seen before in my other classes?

I have seen the arithmetic and relational operators in my math classes.

- What makes a computer science concept fundamental or essential?

A computer science concept is essential or fundamental when it is consistently found in programs. These concepts include variables, conditional statements, loops, functions, and classes.

Notebook responses:

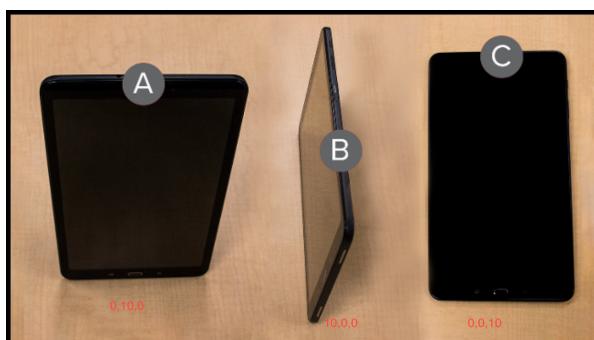
Screen Range:

- Yes. The sprite can be located from 0 to the width of the tablet on the x-axis.
- 0 on the y-axis corresponds to the top of the screen. The width of the tablet as the y coordinate corresponds to the bottom of the tablet.
- Z cannot be changed, because it is physically impossible to make the HappyImage Sprite go off the screen.

Accelerometer Values

- The X and Y values are 0, and the Z value is 10
- The values make sense, because the device is lying flat against the XY plane, so both the X and Y values are 0.

XYZ Accelerometer



- The X and Z values are 0, and the Y value is 10.
- The X value is 10 and the Y and Z values are 0
- The X and Y value are 0, and the Z value is 10

Signature: Franklin Liu	Date: 9/21/21	Cont. on Pg. 6-10
Witness: Ananya Mahadi	Date: 9/21/21	Proprietary Information

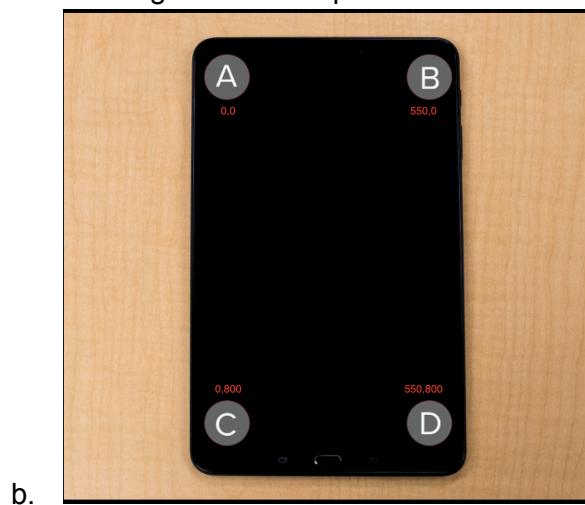
String Formatting

- When I add and remove spaces in different places, there are more spaces in the corresponding places on the screen
- It is possible to add spaces before the Y label by concatenating a string of spaces.

- c. The happyImage sprite's movement remained unchanged.

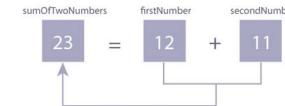
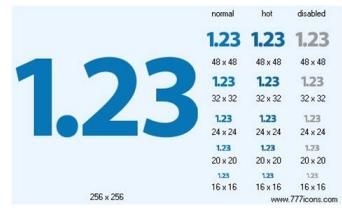
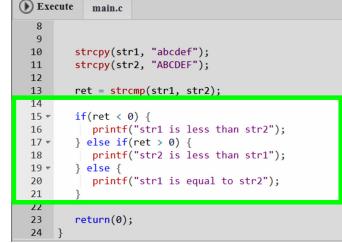
Screen Coordinates

- a. The origin is at the top left corner of the tablet.



b.

Signature: Franklin Liu	Date: 9/21/21	Cont. on Pg. 6-10
Witness: Ananya Mahadi	Date: 9/21/21	Proprietary Information

Term	Example	Meaning	Picture
Arguments	int x = 0; 0 is the argument	A container for data	<pre># include <iostream> using namespace std; int add(int, int); int main() { ... sum = add(num1, num2); // Actual parameters: num1 and num2 } </pre> <pre>int add(int a, int b) { // Formal parameters: a and b ... add = a+b; } </pre>
Variables	int x = 0; x is the variable	The value that a variable contains	<pre>val userName = "Florian" var age = 28 My name is \$userName. I am \$age years old.</pre> 
integer	-1, 0, 1, 2, 3	A whole number that does not have a decimal or any digits after the decimal.	
float	3.14	A type of number that provides very precise information by including all the numbers after the decimal.	
strings	"I like bacon"	Text or characters displayed by a program	
boolean	True	A data type with 2 values: true & false	

Signature: Franklin Liu	Date: 9/21/21	Cont. on Pg. 6-10
Witness: Ananya Mahadi	Date: 9/21/21	Proprietary Information

New vocabulary:

- Algorithms
- Variables
- Arguments
- Strings
 - Concatenation
- Data Types
 - Integer
 - Float
 - Strings
 - Boolean
- Operators
 - Arithmetic operators
 - Logical operators
 - Relational operators
- Calling (a procedure)

New concepts:

- Algorithms, variables, arguments, procedures, strings + concatenations, data types, logic
- Arithmetic operators, relational operators, logical operators
- event-driven programming

Feature list:

1. Show the user the numerical representation of the tablet's motion
2. See the coordinates of the moving object
3. Move the tablet and see motion

Signature: Franklin Liu	Date: 9/21/21	Cont. on Pg. 6-10
Witness: Ananya Mahadi	Date: 9/21/21	Proprietary Information

Screenshots:



Conclusion Questions:

What is an algorithm?

An algorithm is a series of steps that can be repeated to perform a task.

What are some arithmetic operators, relational operators, and logical operators you learned about in this activity?

Several arithmetic operators that I learned about were the addition, subtraction, division, subtraction, and multiplication. Several relational operators I learned about were the greater than ($>$), greater than or equal to (\geq), less than ($<$), less than or equal to (\leq), and the equal to (\equiv) operator. The logical operators that I learned about were conditional statements, whose values were either true or false.

What are some data types that were introduced to you in this activity?

Several data types that I learned about were the integer, string, float and the boolean.

How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I interpreted the Essential Questions from the perspective of someone who is learning about computer science and is relating it to other things they have learned before.

Signature: Franklin Liu	Date: 9/21/21	Cont. on Pg. 6-10
Witness: Ananya Mahadi	Date: 9/21/21	Proprietary Information

Activity 1.1.3: Conditionals and Event-driven Programming: Happy Balance

Essential Questions:

1. Why do you think all decisions in programs are narrowed down to two options, yes or no, true or false?

I think all the decisions in programs are narrowed down to true and false because they are the only two possible values for truth.

2. What information is being hidden or abstracted by the program?

The procedures/functions are being abstracted away by the program.

3. How did you deal with challenges you were confronted with?

I dealt with challenges I was confronted with by continually debugging until I got the satisfactory result.

New vocabulary:

Boolean statements

Chained conditionals

procedure

New concepts:

- Conditionals and Chained Conditionals
- Algorithms, Variables, Arguments, procedures, operators, data types, logic, strings

Feature list:

- Let the player know when they have hit the edge of screen
- Play a sound when they hit the edge of screen
- Allow the player to choose whether to play again

Notebook Responses:

Boolean logic: The two Boolean values that a statement may be evaluated to are true and false.

Procedures: In order to create and use a procedure, one first has to define the procedure and give it name.

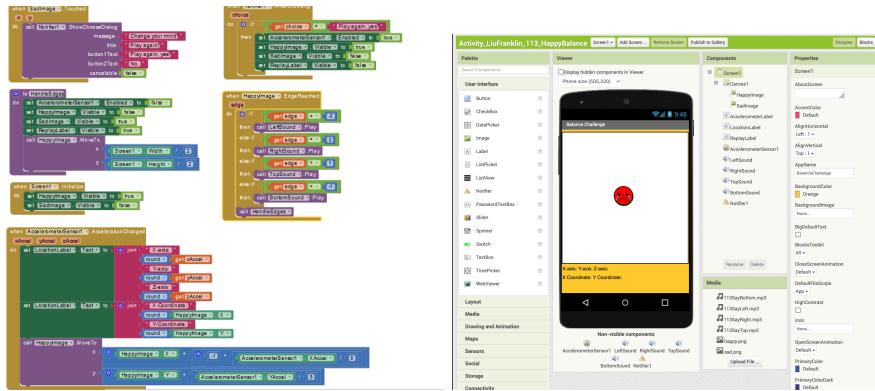
Then they have to specify the steps for the procedure. After this, one has to call the procedure to actually use it.

Signature: Franklin Liu	Date: 9/23/21	Cont. on Pg. 11-13
Witness: Ananya Mahadi	Date: 9/23/21	Proprietary Information

Term	Example	Meaning	Picture																		
Boolean Statements	3 == 3	A statement that evaluates to the values true or false	<p>Writing an or boolean statement:</p>  <p>The outcome will be true as long as one of the expressions evaluates to true. <code>if(x < 10 y > 20)</code> Only one must be true</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>outcome</th> </tr> </thead> <tbody> <tr> <td>true</td> <td>true</td> <td>true</td> </tr> <tr> <td>true</td> <td>false</td> <td>true</td> </tr> <tr> <td>false</td> <td>true</td> <td>true</td> </tr> <tr> <td>false</td> <td>false</td> <td>false</td> </tr> </tbody> </table>	a	b	outcome	true	true	true	true	false	true	false	true	true	false	false	false			
a	b	outcome																			
true	true	true																			
true	false	true																			
false	true	true																			
false	false	false																			
operators	+, -, >=	A symbol that performs an operation on a value. They are used to form boolean statements, which in turn are used in conditional statements	<p>Arithmetic operators</p>  <p>Arithmetic operators are used to perform arithmetic operations. 'C' language supports following arithmetic operators.</p> <table border="1"> <thead> <tr> <th>Operator</th> <th>example</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>+</td> <td>a + b</td> <td>Addition- unary</td> </tr> <tr> <td>-</td> <td>a - b</td> <td>Subtraction- unary</td> </tr> <tr> <td>*</td> <td>a * b</td> <td>Multiplication</td> </tr> <tr> <td>/</td> <td>a / b</td> <td>Division</td> </tr> <tr> <td>%</td> <td>a % b</td> <td>Modulo division- remainder</td> </tr> </tbody> </table>	Operator	example	Meaning	+	a + b	Addition- unary	-	a - b	Subtraction- unary	*	a * b	Multiplication	/	a / b	Division	%	a % b	Modulo division- remainder
Operator	example	Meaning																			
+	a + b	Addition- unary																			
-	a - b	Subtraction- unary																			
*	a * b	Multiplication																			
/	a / b	Division																			
%	a % b	Modulo division- remainder																			
conditionals	<code>if (x > 3): System.out.println("x is greater than 3")</code>	A statement that executes if a boolean statement evaluates to true	<p>Condition is true</p> <pre>int number = 5; if (number > 0) { // code } else { // code } // code after if...else</pre> <p>Condition is false</p> <pre>int number = 5; if (number < 0) { // code } else { // code } // code after if...else</pre>																		

Signature: Franklin Liu	Date: 9/23/21	Cont. on Pg. 11-13
Witness: Ananya Mahadi	Date: 9/23/21	Proprietary Information

Screenshots



Conclusion Questions:

Explain why procedures are considered an abstraction in your program.

The procedures in my program are an abstraction because they can be called multiple times and reused. This allows me to not have to deal with the details of the procedure every time.

Why do you need to initialize or set the values at the beginning of a procedure before using those values in a procedure?

You need to initialize the values at the beginning of a procedure because they need to be defined in order to be usable.

How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I interpreted the Essential Questions from the perspective of someone who is writing a program. I answered the questions with answers that are related to the structure of a program.

Signature: Franklin Liu	Date: 9/23/21	Cont. on Pg. 11-13
Witness: Ananya Mahadi	Date: 9/23/21	Proprietary Information

Activity 1.1.4: Local and Global Variables: Guessing Game - Two Player

Essential Questions:

1. How does the variable scope influence the structure of an algorithm?

Variable scope changes what parts of code access variables with local scope. This way, code becomes more compartmentalized.

2. Why are user stories and user-centered design so important when creating an app?

User stories and user centered design is important because apps need to be designed to be user-friendly and to suit their audience's needs

3. What arithmetic and logical concepts do I keep using over and over?

I constantly use the common arithmetic operators (+, -, /, *) and the AND logical operator.

New vocabulary:

Pseudocode - a substitute for code, used to show the general logic of a program without bothering with correct syntax

Local variables - variables that only certain parts of the code can access

Global variables - variables that any part of the program can access

New concepts:

Local & global variables

Pseudocode & natural language

Signature: Franklin Liu	Date: 9/27/21	Cont. on Pg. 14-18
Witness: Ananya Mahadi	Date: 9/27/21	Proprietary Information

Feature list:

Guess a number

Text box to enter the number guessed

Button to input the number

Receive feedback to adjust their guesses

Feedback label that prompts the user to guess higher or lower

Feedback label that tells the user how many guesses are leftFeedback label that lets the user know whether they won or lost

Control over game conditions

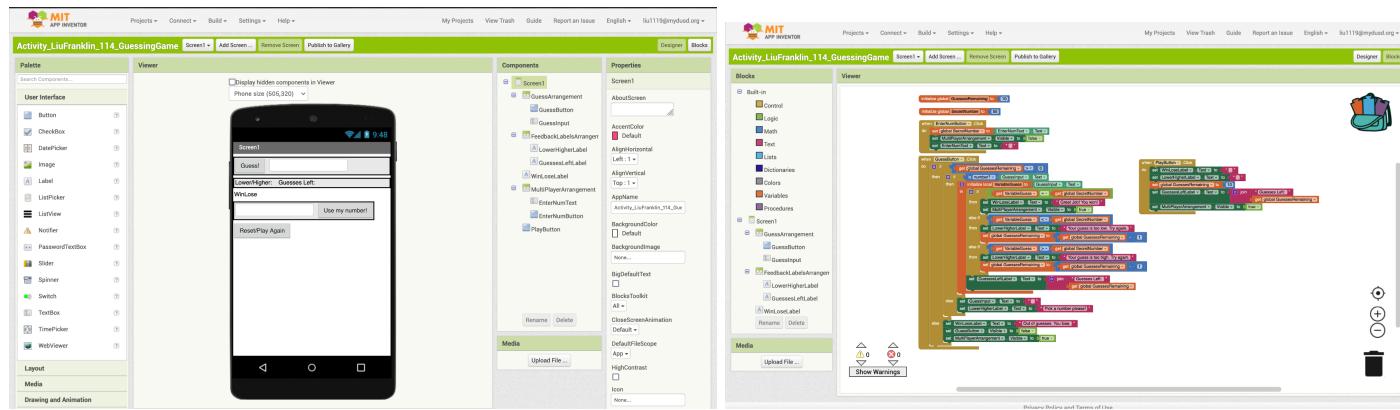
Text box to enter the number the player is trying to guess

Button to input the number the player is trying to guess

A way to limit the number of guesses

Reset/play again button

Screenshots:



Signature: Franklin Liu	Date: 9/27/21	Cont. on Pg. 14-18
Witness: Ananya Mahadi	Date: 9/27/21	Proprietary Information

Notebook Responses:

Design Considerations:

- Split the user interface into sections that different users would use
- Make components easy to read for a user
- Group related components together

Language Abstraction

- Recognize keywords you want to use, and make sure to use them in natural-language descriptions of the code
- In pseudo code, if, then and else are organized like code.
- In natural language, if then and else are written out as full sentences.

Replaying the Game

- In order to replay the game, one would need to make a way for the user to clear all text fields and start again.
- Pseudocode:

```
If PlayButton.Clicked:  
    set WinLoseLabel = "";  
    set LowerHigherLabel = "";  
    set GuessesLeftLabel = "";
```

Global Guesses Placement

You need to add the blocks with the GuessesRemaining variable inside the else if blocks checking if the text input of the GuessInput box is lower or higher than the secret number.

Countdown Variables

The countdown clock should have a local scope, because its only use is inside the Clock1.Timer event handler, and there is no point in allowing other apps to access it.

Signature: Franklin Liu	Date: 9/27/21	Cont. on Pg. 14-18
Witness: Ananya Mahadi	Date: 9/27/21	Proprietary Information

TEMP Chart:

Term	Example	Meaning	Picture
Global variables	<pre>class blah: def words(): global blurb = 3 print(blurb) blah.words() print(blurb)</pre> <p>Output: 3 3 (written in Python)</p>	A variable with a global scope, that can be accessed from any part of the program	<pre> 1 2 3 var global = 10; 4 5 function fun() { 6 7 var local = 5; 8 } 9 10 11 12 13 </pre>
Local variables	<pre>class blah: def words(): blurb = 3 print(blurb) print(blurb)</pre> <p>Output is an error (written in Python)</p>	A variable that cannot be accessed by every part of the program, only by the part of the code where it is defined	<pre> public class SomeClass { public static void SomeMethod(...) { double x; ... int y; } } </pre>

Conclusion Questions:

1. What would happen if a program tried to access a variable that was defined locally in another part of the program?

If a program tried to access a variable that was defined locally in another part of the program, then it would throw up an error, because that variable is not defined in that section of the program.

Signature: Franklin Liu	Date: 9/27/21	Cont. on Pg. 14-18
Witness: Ananya Mahadi	Date: 9/27/21	Proprietary Information

2. How does one of the algorithms in your program function? Use natural language to describe the algorithm as if you were explaining it to your friend who has not taken this course yet.

The algorithm in my program to check the user's guess against the secret number works this way:

- Get the local variable VariableGuess and see if it is equal to the global variable SecretNumber.
 - Then say "Great Job! You Won!"
 - Leave the if statement and move on to the next block of code.
- Or else check if the VariableGuess is less than SecretNumber.
 - Then say "Your guess is too low. Try again."
 - Leave the if statement and move on to the next block of code
- Or else, check if the VariableGuess is greater than SecretNumber.
 - Then say "Your guess is too high. Try again."
 - Leave the if statement and move on to the next block of code

3. How did you interpret and respond to the essential questions? Capture your thoughts for future conversations

I answered the essential questions with the knowledge of the new concepts, from the mindset of a programmer and how the benefits of variable scope help the programmer.

Signature: Franklin Liu	Date: 9/27/21	Cont. on Pg. 14-18
Witness: Ananya Mahadi	Date: 9/27/21	Proprietary Information

Activity 1.1.5: Iteration and Loops: Guessing Game - One Player

Essential Questions:

1. What are the advantages and benefits of using loops in an algorithm?

Using loops in an algorithm is beneficial because it saves work for the programmer. They don't need to write out an instruction again and again; instead, they can just use a loop to write an instruction once.

2. What are the advantages and challenges of pair programming?

The advantages of pair programming are that there is someone else to critique your code and to catch bugs, but some of the challenges that it raises are that the two programmers have to be in constant communication so they can effectively pair program.

3. What are different ways iteration plays a role in a program and in an app that is created for others?

Iteration is used to save work in a program, because it allows a developer to write less code, and it helps organize app development because when programmers use iterative design they build upon previously working iterations in order to create the latest version of an app.

New vocabulary:

Loop - a sequence of instructions that continually repeats until a condition is met

Iteration - A process of repeating a set of instructions a specified number of times or until a condition is met, such as in a repetition of a process or a newer version of development in computer science.

New concepts:

Pair programming

While loops

Incrementing counts

Feature list:

A user input for the number of digits, they want to guess.

They will be able to enter "1" and guess at a number between 0 to 9, or enter "4" and guess at a number between 0000 to 9999.

An algorithm that will generate a random number for the user to guess, based off of a user input for the number of digits they want to guess.

A countdown clock, so the user must guess the correct number before time runs out.

Signature: Franklin Liu	Date: 10/8/21	Cont. on Pg. 19 -21
Witness: Ananya Mahadi	Date: 10/8/21	Proprietary Information

Notebook Responses:

Pair Programming

- Pair programming is practice of programming with two people. One is the Driver, who writes code. The other is the Navigator, who looks at the code, makes suggestions, and catches bugs.
- Successful pair programming teams communicate well and switch often so the roles don't get tedious.

Concatenation of Iterations

The iterations will each generate a random number 0 - 9 and concatenate it to the number, so the number might be 3, then 37, then 371, then 3714, then 3716, then 37167.

Planning Program Flow

The computer should generate a new random number within the PlayButton event handler, because whenever the button to start the game is clicked a new number should be generated to be guessed.

Testing part of Loop

Yes, the testing part of the loop could look like "get count < get NumberOfDigits" because it is essentially the same condition as "get NumberOfDigits > get count"

Counting Iterations

- The loop will go through 6 iterations before it stops
- The number produced by the loop will have six digits at the end, though it may start with a few zeroes.

Design Improvements

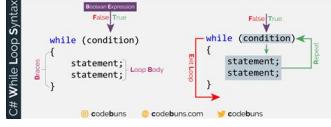
In order to make the game easier to play, I could add more descriptive words on the buttons and labels in order to make it clear what each button does. I could also add a label with a summary of how to play the game.

Screenshots:



Signature: Franklin Liu	Date: 10/8/21	Cont. on Pg. 19 -21
Witness: Ananya Mahadi	Date: 10/8/21	Proprietary Information

TEMP Chart:

Term	Example	Meaning	Picture
loop	While loop	A sequence of instructions that repeats until a condition is true An iteration of a loop is different from an iteration of a project in the way that the same action is repeated every time the loop is iterated through, but new things are added to a project upon every iteration.	

Conclusion Questions:

1. What are two distinct difficulties or opportunities you encountered and how were they resolved or incorporated?

I found it difficult to resolve several bugs in my code, because I find it hard to read block code. I continually read through the code until I spotted the error in order to do the debugging. Additionally, the difficulty of reading the code led to me putting several blocks in the wrong places. I solved this problem by looking at the blocks in question and deciding where they should go.

2. Describe the incremental and iterative development process of your program.

I created the guessing game in 1.1.4, and tested it to make sure it was working correctly. Then I modified the code to make the new single-player version, then tested and debugged again.

3. How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I answered the questions using the new knowledge in this chapter, from the perspective of how iteration benefits a programmer.

Signature: Franklin Liu	Date: 10/8/21	Cont. on Pg. 19 - 21
Witness: Ananya Mahadi	Date: 10/8/21	Proprietary Information

Project 1.1.6: App Development: Creative Expression

Essential Questions:

1. What is the purpose of your program?

The purpose of the program is to provide entertainment. It is a game that allows people to build an ice cream dessert by choosing cones, flavors of ice cream, and toppings.

2. Where does the program integrate mathematical and/or logical concepts?

The program uses basic math to place the ImageSprites of the cone, ice cream and toppings on the Canvas.

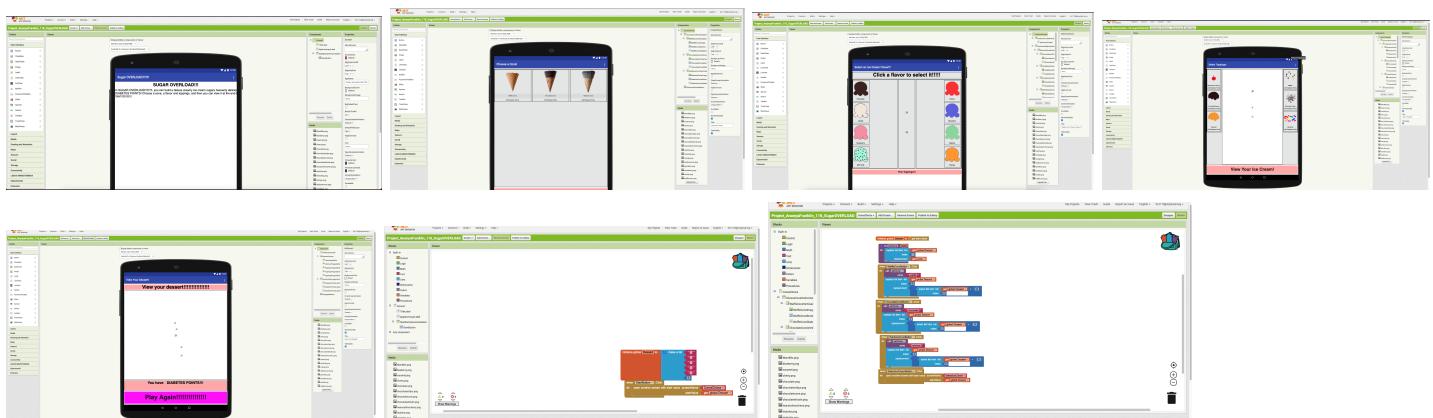
3. What does one of the algorithms do in the program?

One algorithm in the program reconstructs the dessert by iterating through a list that contains the data about each item in the dessert. For example, the list might look like this: "waffle", "chocolate", "chocolatechips", "caramel", "chocolatechips". The first index in the list represents a waffle cone, which the algorithm uses to set an ImageSprite to a picture of a waffle cone. The next index is "chocolate," meaning the user chose a scoop of chocolate ice cream, and another ImageSprite will be set to a picture of chocolate ice cream. The last three indices represent toppings, and these three toppings also get displayed through ImageSprites.

4. How does an abstraction you created manage complexity in the program?

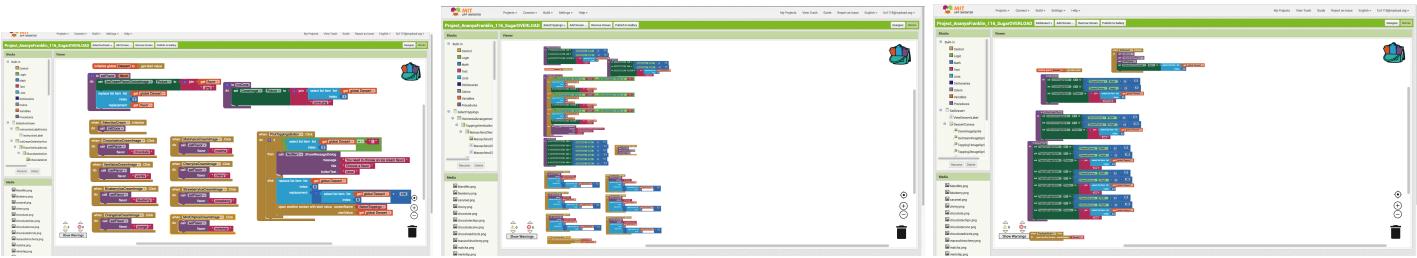
I created procedures to load the images of the elements of the dessert, and to set the flavors of ice cream, toppings, etc. These procedures/functions allowed me to use the same function to set different flavors of ice cream, which saved me lots of work and made the code easier to read.

Screenshots:



Signature: Franklin Liu	Date: 10/28/21	Cont. on Pg. 22-23
Witness: Ananya Mahadi	Date: 10/28/21	Proprietary Information

More Screenshots:



Conclusion Questions:

1. Describe your contribution to the collaboration.

I contributed most of the code, and helped with designing some of the layout.

2. How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I interpreted the Essential Questions from the perspective of a programmer, and I responded to the Essential Questions using details from my program, using in-depth details from the guts of the program.

Signature: Franklin Liu	Date: 10/28/21	Cont. on Pg. 22-23
Witness: Ananya Mahadi	Date: 10/28/21	Proprietary Information

Activity 1.2.1: Problem Solving: Interview Database

Essential Questions:

1. What problems are really worth the effort to try and solve?

Problems that are worth the effort to try and solve with code are problems that can be automated and tedious for a human being to do, or problems that humans are incapable of doing (like multiplying two huge numbers)

2. What does it mean for data to persist?

Data that persists is data that is saved to disk - it stays even after a computer is restarted or an app is closed.

3. What are the similarities and differences between developing an app as creative expression and trying to solve a problem with an app?

Apps developed as a creative expression are designed more with the creator's vision in mind, and are less about the user. Apps that solve problems, however, are more user-centric, and the designer is more interested in what the user wants out of the app. However, both should have an attractive interface and should be easy to use.

New vocabulary:

Database - A storage space on a device used to store information from an app.

Stakeholders - people who will benefit from the app's use

List - a collection of items that can be accessed from a single variable

index - A specific location by order for an individual element in a list.

New concepts:

- Databases, Lists, and Indexes
- Problem Solving
- Professionalism and Engaging Stakeholders in User-centered Design
- Pair Programming
- Algorithms, Variables, Arguments, Procedures, Operators, Data Types, Logic, Loops, and Strings

Signature: Franklin Liu	Date: 11/6/21	Cont. on Pg. 24 - 27
Witness: Ananya Mahadi	Date: 11/8/21	Proprietary Information

Feature list:

- A list of questions to ask in an interview
- The first interview question displayed as soon as the app starts
- A way for a person to speak and fill in the answer to the question
- A way to record interview information
- To automatically display the next question, in the same place, after an answer is recorded
- The app to notify the user when all questions have been asked
- After the last question, the app may be refreshed to the first question
- To store interview information between each app use, so that a user can pull up that information whenever they need it
- A list of all the information stored from interviews
- A way to clear the stored information from previous interviews

Notebook responses:

Interviews

- a. Information that should be off limits for this app are private, sensitive information (like medical information), or irrelevant information about the interviewees' personal life.
- b. I want to be careful about what I capture from the people I interview, because I want only relevant information that will help me research a market, and I don't want to invade my interviewees' privacy.
- c. Interviews should be as concise as possible, and questions should be as direct as possible without offending the interviewee. Also, both the interviewer and interviewee should stay on topic.

Category Questions

I am most interested in technical problems. My five questions for an expert are:

1. How does the reverse engineering process work? (This is because I want to write graphics drivers)
2. Where are most of the pressing technological problems? At the OS level? The user level? The backend? The web?
3. What do most users complain about?
4. What are the most common applications used by everyday people?
5. How would you go about building an app that's easy for technically illiterate people to use?

Signature: Franklin Liu	Date: 11/6/21	Cont. on Pg. 24 - 27
Witness: Ananya Mahadi	Date: 11/8/21	Proprietary Information

Design Components

1. Text fields allow the interviewee to write their name and respond to questions.
2. Labels allow the app to display questions.
3. The speech-to-text component allows the interviewee to answer questions if they don't want to use the keyboard.
4. The ListView component allows the interviewer to see the responses, along with the date and interviewee.

Dated Responses

I would want to have the date of the interview for record-keeping purposes; people's opinions might change over time about the subject I interviewed them on, so it is useful to see how trends and opinions change.

Speech Recognizer

The procedure call makes the speech recognizer record through the microphone and translate the recording into text. Then the text is saved into the result variable, which is then concatenated with what the interviewee has written in the text box to create a final answer. Finally a period is concatenated at the end to make a complete sentence.

User Feedback:

My testers liked the speech-to-text feature, but said the interface could be improved, because the buttons and text fields were grouped in a confusing way.

Interview Questions:

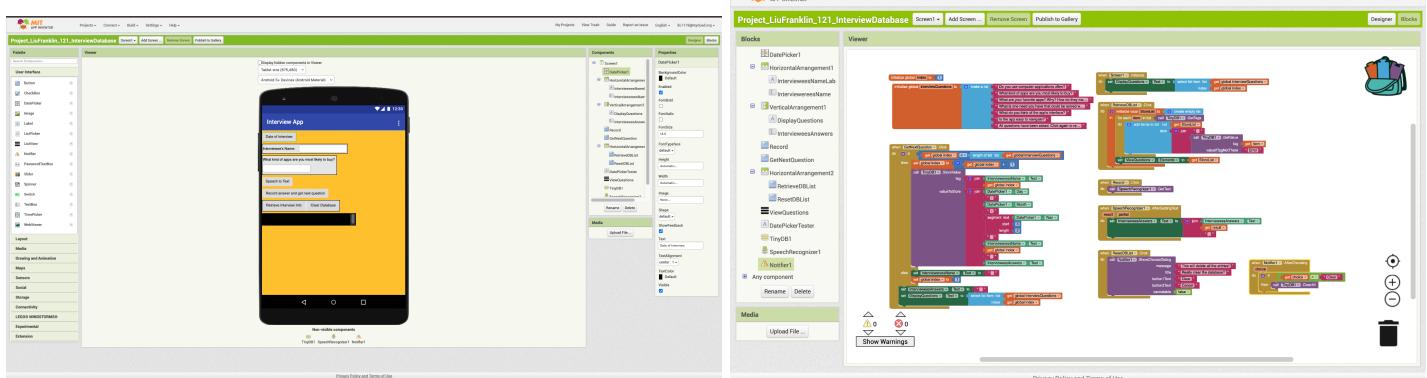
- Do you use computer applications often?
- What is your favorite app? Why? How does it make you feel?
- What is one need you have that could be solved with an app?
- What do you think of the app's interface?
- Is the app easy to navigate?

Potential interviewees:

People I might be able to interview will have free time in their schedule; this might include parents, other students or passers-by. The interview will not take long, so I can just stop and ask them quickly.

Signature: Franklin Liu	Date: 11/6/21	Cont. on Pg. 24 - 27
Witness: Ananya Mahadi	Date: 11/8/21	Proprietary Information

Screenshots:



Conclusion Questions:

1. Why is it so important to engage others in problem identification?

It is important to engage others in problem identification, because different people have different needs and it is almost always impossible to determine what problems other people have without asking them.

2. Why is it so important to get feedback from end users on what you are developing as you are developing it?

Getting feedback from the end users is essential because they allow you to improve your app, and extended testing reveals bugs you may not have noticed.

3. What are three ideas you came across through exploring problem identification and interviewing others that you might pursue later?

First, I came across the idea that user-friendly design is essential, or nobody will use your app. Also, asking other people about problems they need solved is extremely important. Last, problems in almost any field can be solved with code.

4. How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I responded to the Essential Questions from the perspective of a programmer, thinking about how a programmer sees the concepts talked about in the questions.

Signature: Franklin Liu	Date: 11/6/21	Cont. on Pg. 24 - 27
Witness: Ananya Mahadi	Date: 11/8/21	Proprietary Information

Activity 1.2.3: Procedural Abstraction: Price Per Slice

Essential Questions:

1. How are arithmetic and logical concepts integrated into algorithms?

Arithmetic and logical concepts are integrated into algorithms on a deep level. Algorithms use arithmetic to compute values, and they use logical concepts to check conditions. Both of these are important as part of structures like loops and conditional statements.

2. How does abstraction in the programming language make code easier to understand?

Abstraction in programming languages allows the programmer to skip over the more complex or nitty-gritty details. For example, memory management in C (e.g. pointers) is very irritating to deal with, so higher-level programming languages like Python abstract it so the programmer doesn't have to deal with it.

3. How are details being hidden or removed by an abstraction?

Details are hidden to the programmer through abstraction by only revealing an API that the programmer can interact with, which hides the lower level details, like memory management.

New vocabulary

1. Procedural abstraction - The process of applying or using a procedure and only knowing what the procedure does, without knowing how it does it.
2. Abstraction - A technique or process that manages complexity in a program or computer system.
3. Modulo - The remainder from division
4. Decomposition - breaking down a complex problem or system into smaller parts that are more manageable and easier to understand.
5. Procedure - A sequence of actions or instructions to follow in solving a problem or accomplishing a task. Optionally returns a value
6. Function - In programming, a named section of a program that performs a specific task. Returns a value depending on the programming language.
7. Method - A function that is a member of a class. Optionally returns a value

New concepts

- Problem Solving
- Procedural Abstraction
- Programming Language Abstraction: Methods and Functions
- Modulo
- User-centered Design, Iterative Design, and Debugging

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 28-32
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Feature List

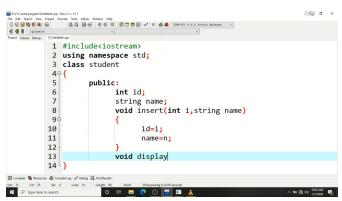
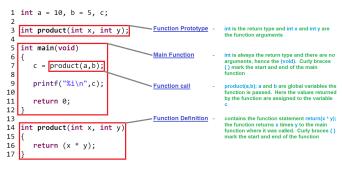
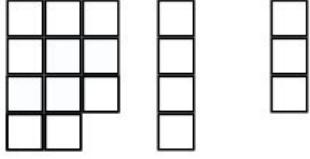
- Allow a user to enter the cost of a whole pizza
- Allow the user to enter the size of a pizza as they would see on a restaurant menu (diameter in inches)
- Calculate the area of a pizza.
- Calculate the cost per square inch pizza for the user.
- Aid the user in counting the number of slices of pizza people want.
- Let the user know how many pizzas to order based on the number of pizza slices they want.
- Let the user know how much to charge people per slice of pizza to cover the cost of buying the pizza.
- Expand upon the app so that the user can compare multiple pizza options all at the same time.

Notebook responses

TEMP Chart:

Term	Example	Meaning	Picture
abstraction	Types of abstraction: Application Software High-level languages Low-level languages Bytes and bits Voltage and current physics	A technique or process that manages complexity in a program or computer system. Is different from algorithms in that it hides information, but algorithms describe a general method to accomplish a path, and the level of detail matters less	
procedures	AreaFormula procedure to calculate the area of a	A sequence of actions or instructions to follow in solving a problem or accomplishing a task. Optionally returns a value	

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 28-32
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

methods	public int add(int a, int b) { return a + b; } (in Java)	A sequence of actions or instructions to follow in solving a problem or accomplishing a task. Optionally returns a value	
functions	def add(a,b): return a+b (in python)	A sequence of actions or instructions to follow in solving a problem or accomplishing a task.	
modulo	13 % 2 = 1	The remainder from division	<p>Modulo operation</p>  $11 \text{ mod } 4 = 3$ <p>ComputerHope.com</p>

Think through a Given Solution

- a. The result for total pizza area seems accurate.
 - i. The number of square inches is usually larger than the diameter input.
 - ii. The number does not seem too large or too small.
- b. The result for the price per area seems accurate
 - i. The price per square inch should be less than the total cost of the pizza
 - ii. The number seems a bit too small.
- c. If there is no input provided, MIT App Inventor throws an error, because it cannot do arithmetic on numbers that don't exist.
 - i. If I enter a diameter but no price, there is an error because an empty string cannot be divided by a number.
 - ii. If I enter the string "ten" instead of 10, there is another error because a string cannot be divided by a number.

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 28-32
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Check the Outputs

- a. The answers I got made sense.
- b. The total pizza area seems accurate
- c. The pizza area per price seems accurate
- d. Other pairs had similar results

App Complexity

Abstraction is helping me manage the complexity by allowing me to ignore low-level details and to focus on the most important part of the algorithm.

Ordering Math

1. I would need at least 8 more slices of pizza
2. The minimum number of slices needed to buy 3 pizzas is 24
 - a. For 8 and 9 slices I would need to order 1 and 2 pizzas, respectively
 - b. For 15, 16 and 17 pizzas I would need to order 2, 2 and 3 pizzas, respectively
 - c.

Number of slices needed	Operation (divided by)	Number of slices per pizza	Output using division operator (quotient)	Output using the round operator	Modulo answer (remainder)
8	\div	8	1	1	0
9	\div	8	1.125	1	1
10	\div	8	1.25	1	2
11	\div	8	1.375	1	3
12	\div	8	1.5	2	4
13	\div	8	1.625	2	5
14	\div	8	1.75	2	6
15	\div	8	1.875	2	7
16	\div	8	2	2	0
17	\div	8	2.125	2	1

- d. The division operator can generate a float. The round operator converts floats to integers.
- e. 320 slices should have 40 pizzas. 321 slices should have 41 pizzas.

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 28-32
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Price Expansion

If I used Price1Input instead of the PizzaPrice box, I would changed the OutPut method to use the Price1Input value instead of the PricePerSlice value.

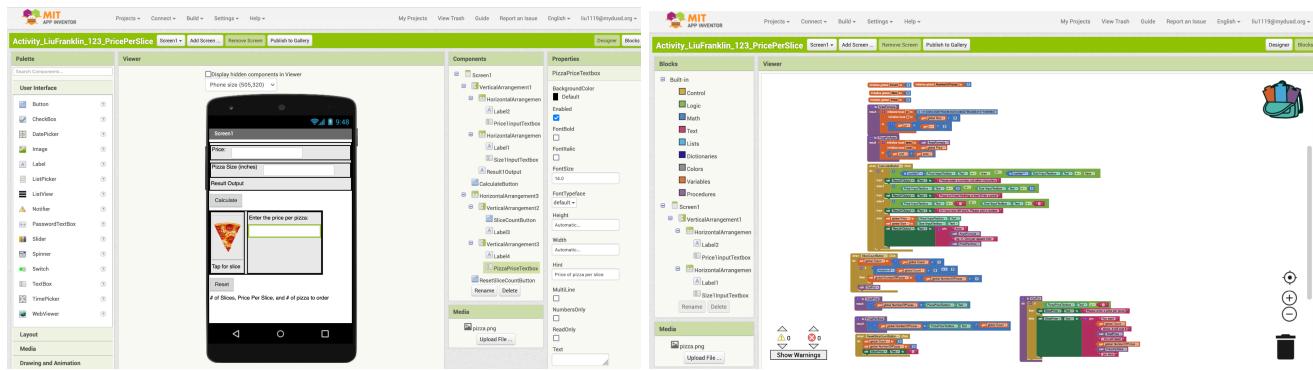
Procedure Planning

My procedure multiplies the number of pizzas by the price of pizzas, which is found in the pizza price textbox. This abstracts this process, because it uses variables and not specific numbers, which allows me to focus on the general process.

Planning Procedural Abstraction

The procedure to get the price per slice divides the total pizza price by the number of slices, which abstracts the process by using variables, and not with specific numbers, which is less work.

Screenshots



Conclusion Questions

1. In your own words, describe what modulo is doing in your program.

I use modulo to check how many whole pizzas the user wants to order, and how many individual slices they want, which is then used to calculate the amount they have to pay and the price per slice.

2. How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I interpreted and responded to the Essential using knowledge about abstraction and modular arithmetic, and I focused on what algorithms look like to a programmer.

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 28-32
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Activity 1.2.4: Lists: Survey Says

Essential Questions

1. Why are lists considered essential in computer science?

Lists are useful in computer science because they allow you to store multiple values that are related.

2. Why is sharing code and looking at many examples important to people writing programs?

Sharing code and looking at examples is important to programmers because it allows them to build on the work of others and write better code

3. How have you gotten better at collaborating with your partners when pair programming?

Pair programming has helped my collaboration skills.

New vocabulary

- List - a data structure containing multiple values accessible by their indices
- Best So Far Loops - loops that iterate through a list and get the best value
- Accumulators - variable that keeps a running or cumulative total or value. Commonly an accumulator is initialized outside of a loop, assigned new values within the loop, and then its result is used outside of the loop.
- New concepts
- Lists and Indexes
- Best So Far Loops
- Accumulators
- Algorithms, Variables, Arguments, Procedures, Operators, Data Types, Logic, Loops, and Strings

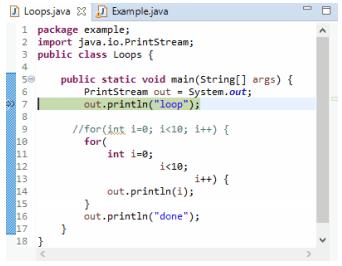
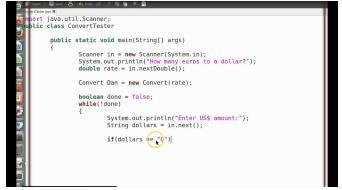
Feature List

The user needs the ability to:

- Have a custom background color on the user interface.
- Have the polling list started with a hard coded answer to the polling questions
- Select a response from a list of polling responses.
- Type in a response of their favorite ice cream flavor that is not already in the list.
- See a list of all the other polling responses.
- Vote for a favorite item in the list (if it was already populated by another user).
- Confirm a vote for an item in the list.
- Replace an item in the list if it is misspelled.
- Display the number of votes an item in the list received.
- Display the item receiving the most votes.
- Display the total number of votes made in the poll.

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 33-35
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Notebook Responses
TEMP Chart:

Term	Example	Meaning	Picture
Lists	[1, 2, 3] List in python	Lists are a way to store information in a collection that is related.	
Best-So-Far loop	Mylist = [0, 1, 2, 3] Biggest_number = 0 For i in MyList: If i > biggest_number: Biggest_number = i	A kind of loop that iterates through a list and finds the “best” value	
Accumulator	accumulator = 0 For item in list: Accumulator += item	A role that a variable may take on in which it keeps a running or cumulative total or value. Commonly an accumulator is initialized outside of a loop, assigned new values within the loop, and then its result is used outside of the loop. This is why it is better to have accumulators as a local variable	

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 33-35
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information

Planning for User Inputs

I can sanitize user input before entering it into a system. Additionally, I can use conditional statements to check if user input is the wrong type or if it does not exist.

Incrementing Across Lists

Separate lists may contain related information because they can contain different aspects of the same thing. For example, one list can store people's names, while another can store their ages.

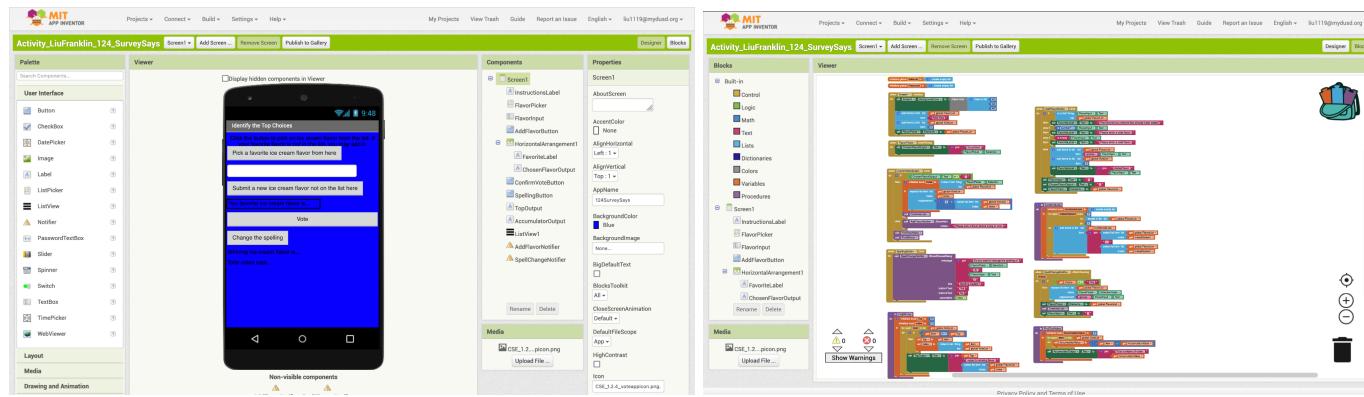
Combine Lists Procedure

The CombineLists procedure iterates through two lists of the same length and concatenates their outputs. The ListView1 update to CombinedLists could possibly be put in the AddFlavorButton.Click event handler.

Code Review

- a. The AddFlavorButton checks if input is valid and then adds them to a list of flavors and a list of possible flavors to vote for. Then it clears various text fields so the user is not confused.
- b. It is important to know and use the index values in a list because they allow you to keep track of your position in a list
- c. The for loop and the best so far loop look through the data, with a starting value of the "best" value in a list and checking if each successive value in the list is better than the current "best" value
- d. Procedural abstraction was useful because it allowed me to focus on the qualities of each value in the list without actually looking at the values

Screenshots



Conclusion Questions

1. Describe the difference between incrementing a count and an accumulator.

An accumulator is a running total, while a count is a fixed number.

2. How did you interpret and respond to the Essential Questions? Capture your thoughts for future conversations.

I responded to the Essential Questions from the perspective of a programmer and using my new knowledge about lists and loops.

Signature: Franklin Liu	Date: 12/2/21	Cont. on Pg. 33-35
Witness: Ananya Mahadi	Date: 12/2/21	Proprietary Information