CoolThink@JC –
Computational Thinking Education Programme

Course Code & Title

CF3E Computational Thinking Development - Level 3

Unit Code & Topic

L3U8.4-8.5
Factor App

香港教育大學 The Education University of Hong Kong

Massachusetts Institute of Technology

Funded by

香港賽馬會慈善信託基金 The Hong Kong Jockey Club Charities Trust 同心同步同進 RIDING HIGH TOGETHER

These teaching and learning materials were co-created by The Education University of Hong Kong, and Massachusetts Institute of Technology, and funded by The Hong Kong Jockey Club Charities Trust. (Revised August 2019)

Table of Contents

1.	Synopsis	1
2.	Unit Intended Learning Outcomes (UILOs)	1
3.	Mapping with the Computational Thinking Framework	2
4.	Learning Prerequisite	4
5.	Pedagogical Approach	4
6.	Lesson Plan	4
7.	Instructional Details	9
8.	Assessment	10
9.	Screen Design and Code	13
Appendix 1	L3U8.4-8.5 Teacher's Guide: Lesson 1	17
Appendix 2	L3U8.4-8.5 Teacher's Guide: Lesson 2	20
Appendix 3	L3U8.4-8.5 Teacher's Guide: Lesson 3	23
Appendix 4	L3U8.4-8.5 Teacher's Guide: Lesson 4	25

1. Synopsis

Students will develop a math app in this unit, and focus on the computational thinking concept of repetition by learning to use a for loop to make a list of factors for a number. The app will help students to practice their math skills by testing their knowledge of prime and composite numbers and their factors. They will learn the remainder procedure in order to determine factors of numbers, and will learn an algorithm for determining if a number is prime. They will get more practice writing procedures to abstract and modularize when coding. The focus for Lesson 1 and 2 is algorithmic thinking. The focus for Lesson 3 is repetition. The focus for Lesson 4 is conditionals.

2.Unit Intended Learning Outcomes (UILOs)

After completing this unit, students will be able to:

UILO$_1$: create a factor math app;

UILO$_2$: demonstrate understanding of the remainder procedure and its role in determining the factors of a number;

UILO$_3$: convert steps in an algorithm to code blocks;

UILO$_4$: apply the computational thinking practices of abstracting and modularizing, and being incremental and iterative;

UILO$_5$: develop their computational thinking perspectives of expressing, connecting, and questioning by creating an app to solve an everyday mathematical problem;

UILO$_6$: reinforce their understanding of the computational thinking concept of repetition by correctly using a for loop; and

UILO$_7$: reinforce their understanding of conditionals to test for a correct answer in an app.

3. Mapping with the Computational Thinking Framework

The following tables show the alignment of this unit with the intended learning outcomes of the computational thinking framework. The entries indicate the expected relevance of this unit to each outcome:

✓✓✓	:	High relevance
✓✓	:	Some relevance
✓	:	Low relevance

Computational Thinking Concepts

L3U8.4-8.5: Factor App		
1. Sequences	✓✓	The order of code blocks matters when finding factors of numbers.
2. Events	✓	Button click events are part of this app.
3. Repetition	Lesson 3 focus	A for loop is used to repeatedly divide a number to find its factors.
4. Conditionals	Lesson 4 focus	Conditionals are used to check for correct answer in prime and composite button click events.
5. Parallelism		
6. Naming	✓✓	Naming of procedures and the variables in for loop are important.
7. Operators	✓✓✓	Logic equals and greater than blocks are used in conditionals.
8. Manipulation of data and elementary data structures	✓✓✓	A list is used to store factors of a number, numbers are added to the list, and length of list is used to determine the number of factors.

Computational Thinking Practices

L3U8.4-8.5: Factor App		
1. Reusing and remixing	✓✓	Student reuse PrimeButton.Click block to create CompositeButton.Click.
2. Being incremental and iterative	✓✓	Students create the app in stages, testing and debugging as they go.
3. Abstracting and modularizing	✓✓✓	Students write “findFactors” procedure to find the factors of a number.
4. Testing and debugging	✓✓	Students test the app to make sure the app works.
5. Algorithmic thinking	Lesson 1&2 focus	Students apply an algorithm in their code to find factors of a number to determine if the number is prime or composite.

Computational Thinking Perspectives

L3U8.4-8.5: Factor App		
1. Expressing	✓	Students learn that math problems can be solved creatively by making an app.
2. Connecting	✓✓	Students can share the factor app with their friends and family.
3. Questioning	✓✓	Students learn how computational thinking can be applied to solve complex mathematical problems, and how they can connect technology to problems they encounter.
4. Computational identity	✓✓	The factor app relates to their work in school, by helping them better understand prime and composite numbers.
5. Digital empowerment	✓✓	Students are empowered to build an app to help themselves and others.

4. Learning Prerequisite

Students should have completed Level 1 and 2 of the CoolThink@JC curriculum and the first two units of Level 3. Students should also have a basic understanding of prime and composite numbers.

5. Pedagogical Approach

The suggested pedagogical approach of this unit is as follows:

- Class discussion;
- Classroom demonstration;
- Guided individual coding.

6. Lesson Plan

This unit consists of 4 lessons of 35 minutes.

Lesson 1

Time	Activity
5 mins	**Introduction to Unit** 1. Ask students: (1) Do you recall the Addition App made in Level 2? (2) Do you remember how the app worked? 2. Explain to students they will create a new math app to help with practice for deciding if a number is prime or not.
10 mins	**Hands-on** 1. Let students play the completed Factor App. (FactorApp_Completed.aia) 2. Ask students to complete *Student Guide: Lesson 1*. 3. Review their answers in the student guide.
15 mins	**Prime and Composite Number Discussion** 1. Ask students: (1) "What is a factor?" (Divides into another number exactly and without leaving a remainder.) (2) "What is a prime number?" (Only 2 factors: 1 and the number itself, e.g. 7) (3) "What is a composite number?" (More than 2 factors, e.g. 8) 2. Ask students what will happen if they input "1" or "0" into the app. 3. Have students try to input 2 prime numbers, 569 and 991, then input a large number (563879) into the app and see what will happen. Warn students not to put in a larger number, because it could cause the app to stop. 4. Questioning: Students have to wait 2-3 minutes for the outcomes, during which time they can discuss why they think it might take so long to figure out whether it is prime. What do they think the app is doing?
5 mins	**Wrap-up** 1. Review prime and composite numbers and the Factor App. 2. Explain to students that they will be coding their own Factor app and learning an algorithm to determine whether a number is prime or composite.
Post-Lesson Activities	(Required): Complete *Student Guide: Lesson 1* if not finished in class.

Lesson 2

Time	Activity
10 mins	**Simple Remainder App** 1. Introduce the remainder block in MIT App Inventor with the student guide. 2. Ask students to follow *Student Guide: Lesson 2* to code this app.
10 mins	**Algorithmic Thinking** 1. Ask students to complete page 4-5 of *Student Guide: Lesson 2*. 2. Review the algorithm (finding factors through repetition) with students.
10 mins	**Display Number App** 1. Teacher reviews the for each number code block with students to explain how the "for each number" block works. 2. Ask students to follow *Student Guide: Lesson 2* to update the code for the DisplayNumber app.
5 mins	**Wrap-up** 1. Review the usage of remainder and for each number blocks. 2. Tell students that they will make use of remainder and for blocks to program the "Factor App" in the next lesson.

Lesson 3

Time	Activity
5 mins	**Introduction to Factor App** 1. Ask students: "What are the functions of the Factor App?" (Show factors of a number and check if it is prime or composite.) 2. Ask students to fill in the blanks on page 1 of *Student Guide:Lesson3*. 3. Go over the answers and review the flow and logic of the Factor App.
25 mins	**Coding Activity** 1. Review the algorithm from Lesson 2 to find factors for a number. (1) Count up from 1 to the Number. If remainder of (Number ÷ Divisor) = 0, then Divisor is a factor of Number. 2. Explain to students they will code a procedure called "findFactors" to create a list of factors for a number. 3. Ask students to follow *Student Guide: Lesson 3* to code the procedure.
5 mins	**Wrap-up** 1. Review the procedure "findFactors". 2. Tell students that they will code the two buttons, PrimeButton and CompositeButton in the next lesson.

Lesson 4

Time	Activity
10 mins	**Review Prime and Composite Numbers** 1. Ask students: (1) "Can you tell if 7 is a prime or a composite number?" (Prime) (2) "Why is 7 prime?" (Factors: 1 and 7 itself.) 2. Explain that the Factor App will calculate the number of factors of a number to determine if a number is prime. It counts how many factors it found in "findFactors", and if the only factors are 1 and the number itself (i.e., only 2 factors), it is prime. 3. On the other hand, if the number of factors of the input number is more than 2, it is composite. 4. Ask students what conditionals they think they should use for PrimeButton and CompositeButton Click events.
20 mins	**Coding Activity** 1. Ask students to follow *Student Guide: Lesson 4* to complete the app. 2. (Optional) Have students to try *Student Guide: Challenge* to account for users entering 0 in the app.
5 mins	**Wrap-up** 1. Review the computational thinking concepts learnt in this unit. 2. Review the computational thinking practices used in this unit.
Post-Lesson Activities	1. (Required): Complete multiple choice questions and survey of learning attitudes. 2. (Optional): Add other features such as scoring and a clock timer.

7. Instructional Details

The teaching and learning materials of the four lessons are presented in the appendices below. Please note the inclusion of the template and checkpoint aia files. These MIT App Inventor project files are included as backup resources for teachers. Each checkpoint aia file includes the project and code blocks that are expected to be completed during that lesson. If necessary, teachers can use these as reference or even a start project file for students who miss a lesson due to illness.

Lesson 1

1. Teacher's Guide: Lesson 1
2. Student Guide: Lesson 1
3. FactorApp_Completed.aia file

Lesson 2

1. Teacher's Guide: Lesson 2
2. Student Guide: Lesson 2
3. SimpleRemainder_Template.aia file
4. SimpleRemainder_Completed.aia file
5. DisplayNumber_Template.aia file
6. DisplayNumber_Completed.aia file

Lesson 3

1. Teacher's Guide: Lesson 3
2. Student Guide: Lesson 3
3. FactorApp_Template.aia file

Lesson 4

1. Teacher's Guide: Lesson 4
2. Student Guide: Lesson 4
3. FactorApp_Checkpoint1.aia file
4. Student Guide: Challenge
5. FactorApp_Challenge.aia file

8. Assessment

The following table maps the assessment modes and corresponding Unit Intended Learning Outcomes (UILOs) in Section 2.

Assessment Modes	UILO
Self-assessment: 1. Online multiple-choice questions 2. Survey of learning attitudes	$UILO_{2,5,6}$

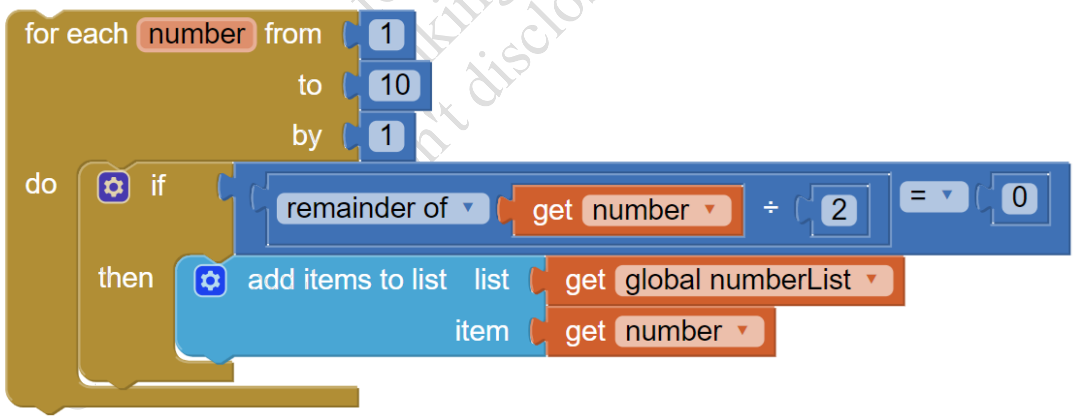
Self-assessment

Self-assessment allows students to reflect on their learning performance. There are two sorts of self-assessment questions below:

Online multiple-choice questions

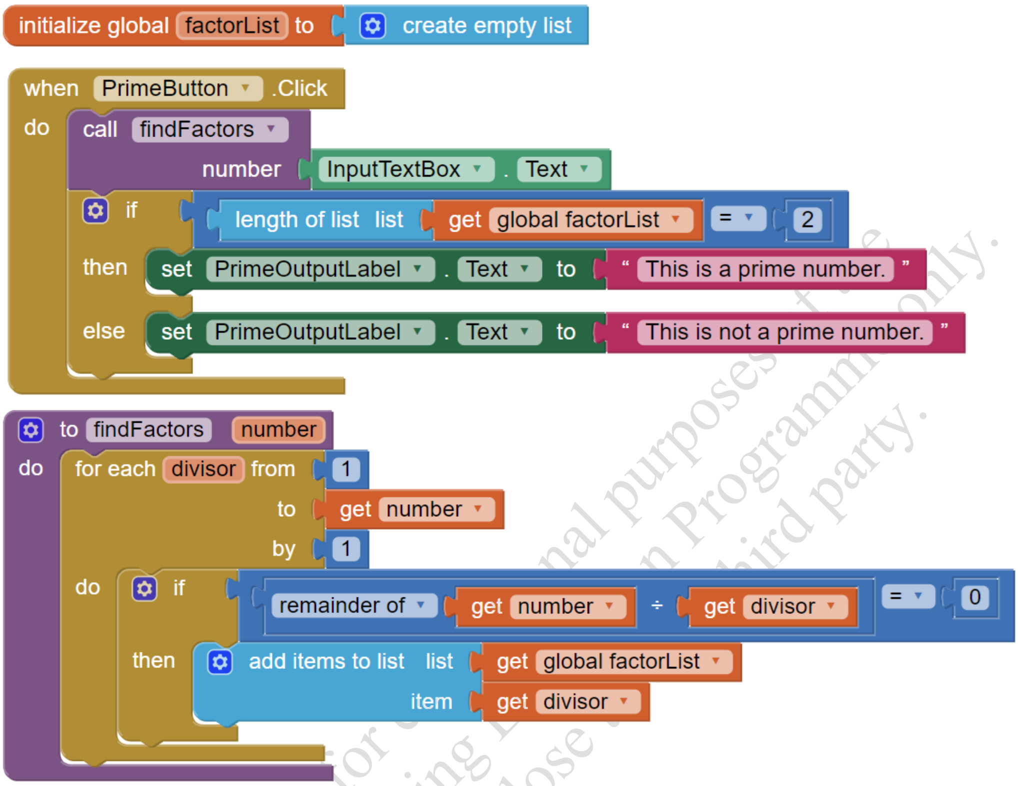
Online multiple-choice questions are adopted to assess students' understanding of the key concepts of the activity, as shown below.

1. After the following code block is executed, what is the value of numberList?



A. (1 2 3 4 5 6 7 8 9 10)
B. (2 4 6 8 10)
C. (1 3 5 7 9)
D. empty list
(Answer: B)

2. Peter uses the following code for his app. He types in the number 7 into InputTextBox, and then presses the PrimeButton. The PrimeOutputLabel says "This is a prime number". He presses PrimeButton a second time. The PrimeOutputLabel says "This is not a prime number. What is wrong?



A. He needs to type 7 again.

B. He should call the "findFactors" procedure after the if blocks.

C. He needs to set global factorList to an empty list at the start of the "findFactors" procedure.

D. The if statement with the remainder block should test if it is equal to 1, not 0.

(Answer: C)

Survey of learning attitudes

To evaluate students' attitudes, perceptions, and understandings of coding, they are required to complete the following questionnaire by putting a "✓" in the appropriate boxes.

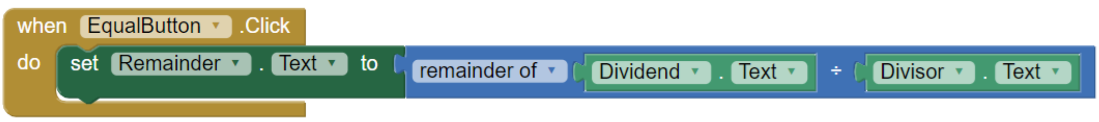
After completion of this unit, I think…	Disagree	Somewhat disagree	Neutral	Somewhat agree	Agree
I feel inspired when learning more about programming.					
After learning programming, I have a deeper understanding of how digital technology works.					
Programming helps me relate to people in my community.					
The learning activities of this unit are interesting and I like them.					
I am excited to share this app with friends and family.					

9. Screen Design and Code

Simple Remainder App (SimpleRemainder_Completed.aia file)

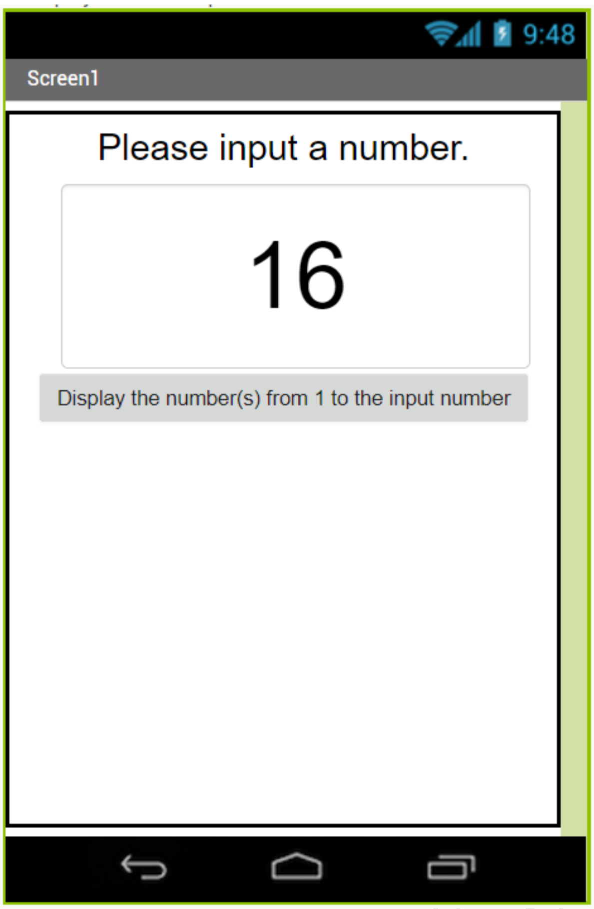
Designer

Blocks

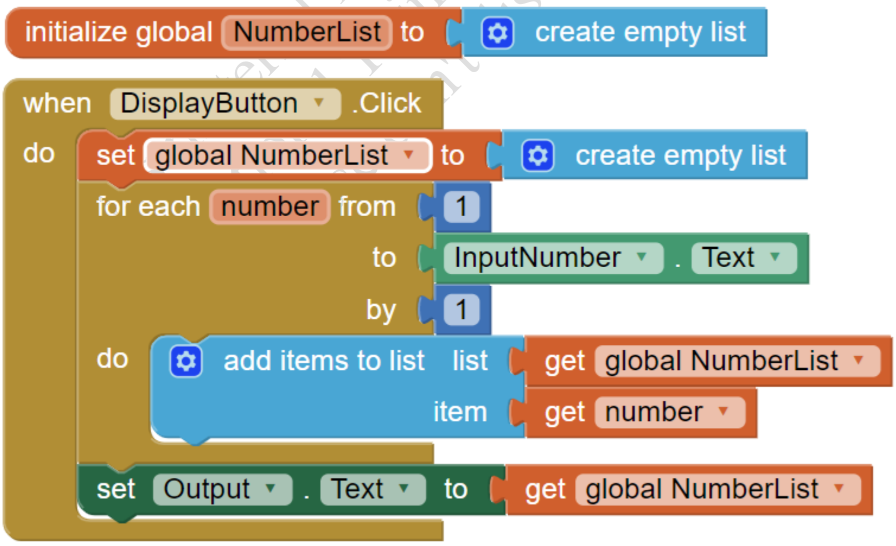


Display Number App (DisplayNumber_Completed.aia file)

Designer

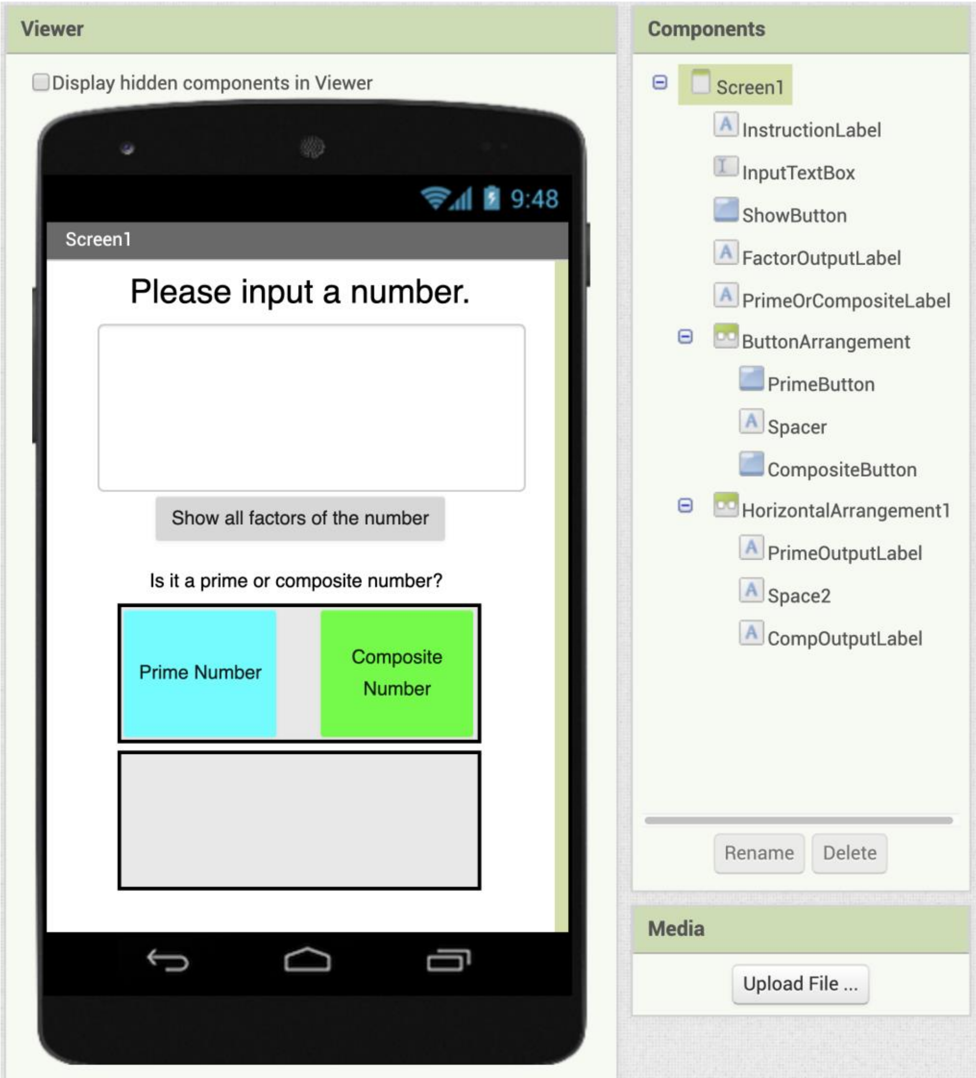


Blocks

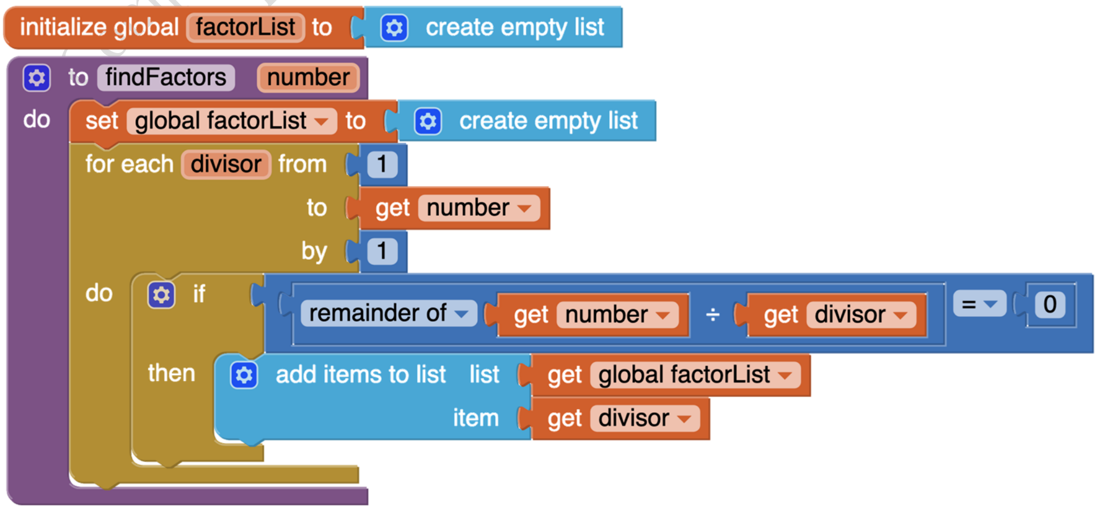


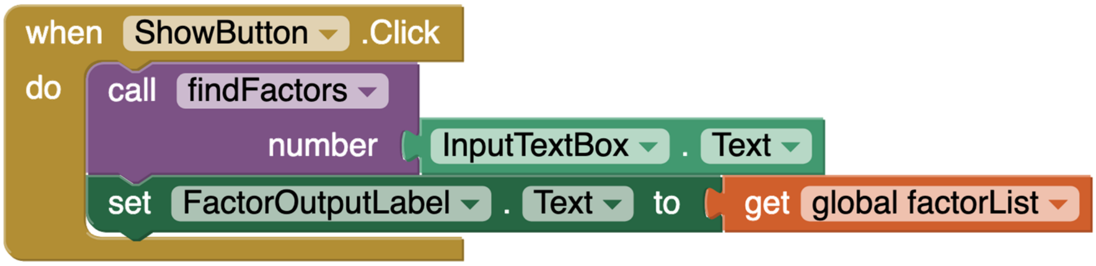
Factor App (FactorApp_Completed.aia file)

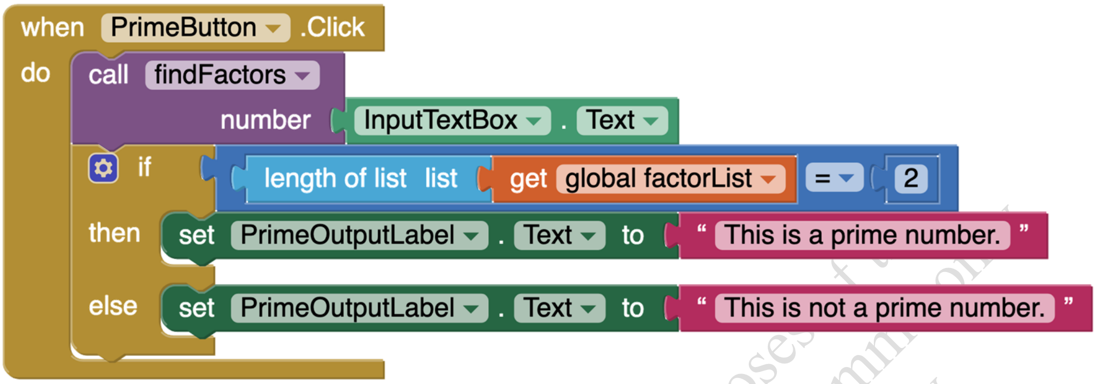
Designer

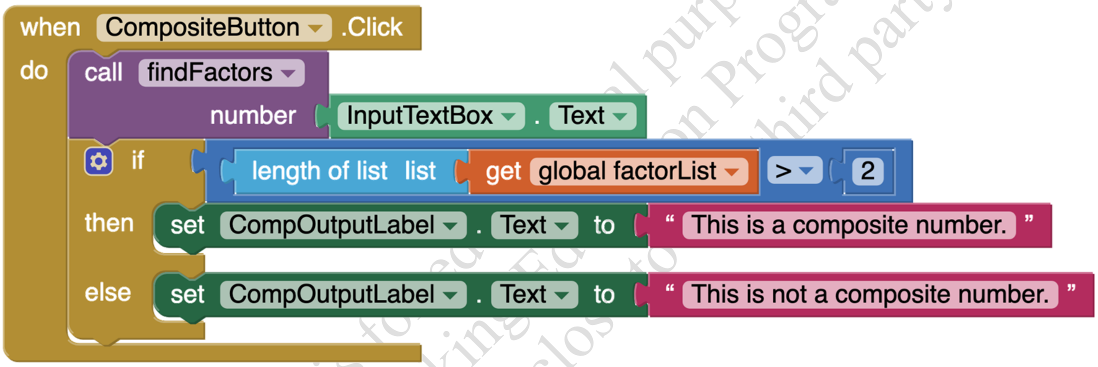


Blocks

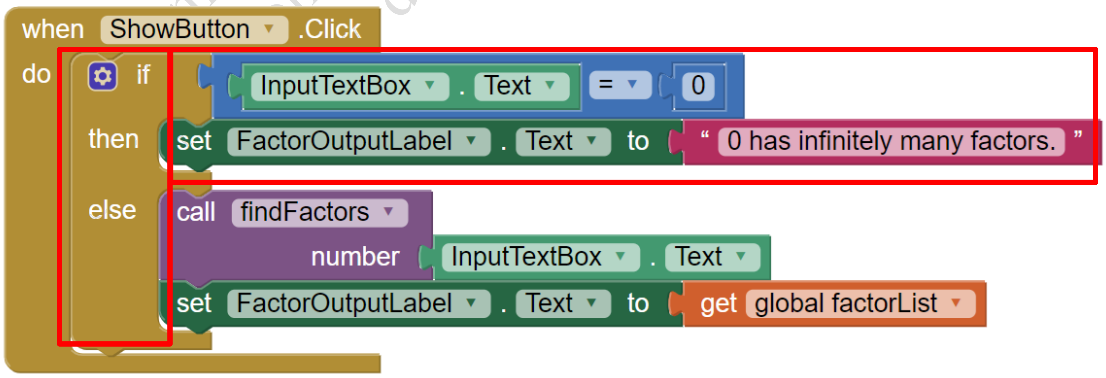








Challenge: (FactorApp_Challenge.aia file)



Appendix 1
L3U8.4-8.5 Teacher's Guide: Lesson 1

Teaching Objectives

1. Teach students about prime and composite numbers, and how to determine factors of a number.
2. Focus: Develop students' algorithmic thinking skills by finding factors for a number in a systematic way.

Lesson Outline

Introduction to Unit (5 minutes)

In this introduction, make a connection for the students to the work they did in Level 1 and Level 2 of the curriculum, and how this unit is a continuation of relating programming to everyday life.

1. Ask students:

   (1) Do you recall the Addition App made in Level 2?

   (2) Do you remember how the app worked?

2. Explain to students they will create a new math app to help with practice for deciding if a number is prime or not.

Hands-on (10 minutes)

1. Let students play the completed Factor App. (FactorApp_Completed.aia)
2. Ask students to complete *Student Guide: Lesson 1*.
3. Review the answers in the student guide:

   (1) Show all factors of the number button (**ShowButton**):

   a. Find the factors of the inputted number.

   b. Show the factors of the inputted number in the **FactorOutputLabel**.

   (2) Prime Number button(PrimeButton):

   a. Find the factors of the inputted number.

   b. Check if number of factors = ___2___ , correct.

   c. Show the result in the PrimeOutputLabel.

   (3) Composite Number button (CompositeButton):

   a. Find the _factors_ of the inputted number.

   b. Check if number of factors > __2___ , correct.

   c. Show the result in the CompOutputLabel.

Equation	Divisor	Remainder	Is Factor?
7 ÷ 1	1	0	✓
7 ÷ 2	2	1	
7 ÷ 3	3	1	
7 ÷ 4	4	3	
7 ÷ 5	5	2	
7 ÷ 6	6	1	
7 ÷ 7	7	0	✓

Equation	Divisor	Remainder	Is Factor?
8 ÷ 1	1	0	✓
8 ÷ 2	2	0	✓
8 ÷ 3	3	2	
8 ÷ 4	4	0	✓
8 ÷ 5	5	3	
8 ÷ 6	6	2	
8 ÷ 7	7	1	
8 ÷ 8	8	0	✓

Prime and Composite Number Discussion (15 minutes)

1. Ask students:
   (1) "What is a factor?" (Divides into another number exactly and without leaving a remainder.)
   (2) "What is a prime number?" (Only 2 factors: 1 and the number itself, e.g. 7)
   (3) "What is a composite number?" (More than 2 factors, e.g. 8)
2. Ask students what will happen if they input "1" or "0" into the app.
3. Have students try to input 2 prime numbers, 569 and 991, then input a large number (563879) into the app and see what will happen. Warn students not to put in a larger number, because it could cause the app to stop.
4. Questioning: Students have to wait 2-3 mins for the outcomes, during which time they can discuss why they think it might take so long to figure out whether it is prime. What do they think the app is doing? (*It is looking for all the factors, so it needs to count from 1-563879).*

Wrap-up (5 minutes)

1. Review prime and composite numbers and the Factor app.
2. Explain to students that they will code their own Factor app and learning an algorithm to determine whether a number is prime or composite.

Post-Lesson Activities

(Required) Complete Student Guide: Lesson 1 if not finished in class.

Appendix 2
L3U8.4-8.5 Teacher's Guide: Lesson 2

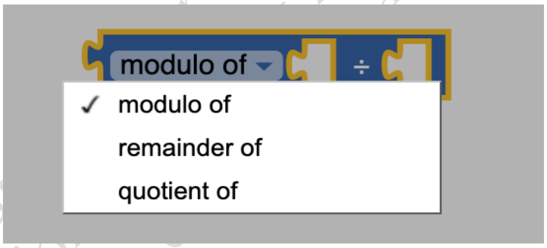
Teaching Objectives

1. Teach students the remainder block in MIT App Inventor.
2. Teach students about repetition with the for each number block in MIT App Inventor.
3. Focus: develop students' algorithmic thinking skills by creating an algorithm for determining if a number is prime.

Lesson Outline

Simple Remainder app (10 minutes)

1. Introduce the remainder block in MIT App Inventor with the student guide. The remainder block is found in the Math drawer, as a dropdown on the modulo block.



2. Ask students to follow *Student Guide: Lesson 2* to code this app. Students should try various numbers in the app to see that it returns the correct remainder in the app.

Algorithmic Thinking (10 minutes)

1. Ask students to complete page 4-5 of *Student Guide: Lesson2*.

Number	Divisor	Remainder	Factor?	Algorithmic Thinking
8	1	0	✓	E.g. If remainder (Number÷1) = 0, then 1 is a factor of the number.
8	2	0	✓	If remainder (Number÷2) = 0, then 2 is a factor of the number.
8	3	2		If remainder (Number÷3) = 0, then 3 is a factor of the number.
8	4	0	✓	If remainder (Number÷4) = 0, then 4 is a factor of the number.
8	5	3		If remainder (Number÷5) = 0, then 5 is a factor of the number.
8	6	2		If remainder (Number÷6) = 0, then 6 is a factor of the number.
8	7	1		If remainder (Number÷7) = 0, then 7 is a factor of the number.
8	8	0	✓	If remainder (Number÷8) = 0, then 8 is a factor of the number.

Number	Divisor	Remainder	Factor?	Algorithmic Thinking
8	1	0	✓	E.g. If remainder (Number÷Divisor) = 0, then Divisor is a factor of the number.
8	2	0	✓	If remainder (Number÷Divisor) = 0, then Divisor is a factor of the number.
8	3	2		If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.
8	4	0	✓	If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.
8	5	3		If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.
8	6	2		If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.
8	7	1		If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.
8	8	0	✓	If remainder (Number÷ Divisor) = 0, then Divisor is a factor of the number.

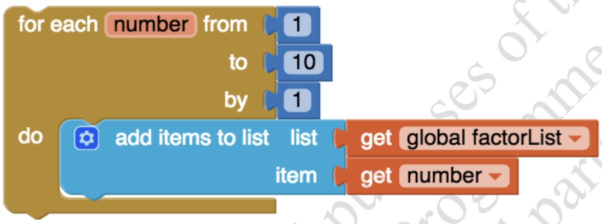
2. Review the algorithm (finding factors through repetition) with students.

   Counting up from 1 to the **Number**:

   If remainder of (Number ÷ Divisor) = 0, then Divisor is a factor of Number.

Display Number App (10 minutes)

1. Teacher reviews the code blocks with students to explain how the for each number block works. Briefly demonstrate the for loop in MIT App Inventor. The “for each number” block, found in the Control drawer, repeats the inner code based on its input parameters. The “number” parameter is initialized to the “from” parameter, in the case below, 1. The inner code is executed, with “number” holding that value. After the loop executes, “number” is increased by the “by” parameter, and the code block is repeated, this case number = 2. This continues until number equals the “to” parameter, in this case 10. The code block is executed, then the for loop is exited. The code block below is repeated 10 times, with number holding the value 1,2,3, ... 10.



2. Ask students to follow *Student Guide: Lesson 2* to update the code for the DisplayNumber app. Students change the “for each number” block to use the InputNumber.Text block instead of a number (5).
3. Students play the app with different number values. Try 0. Try 1.

Wrap up (5 minutes)

1. Review the usage of remainder and for each number blocks.
2. Tell students that they will make use of remainder and for blocks to program the Factor App in the next lesson.

Appendix 3
L3U8.4-8.5 Teacher's Guide: Lesson 3

Teaching Objectives

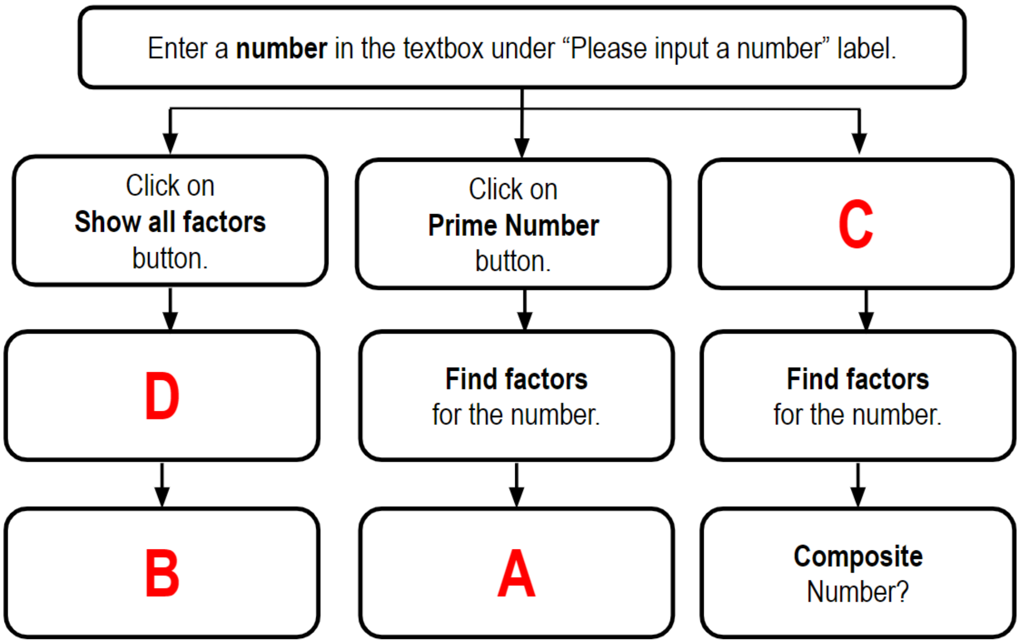
1. Reinforce students' use of abstraction and modularization by showing them how to create a "findFactors" procedure.
2. Focus: Help students to use repetition with the for each number block in their find factors algorithm.
3. Reinforce the use of conditionals in an app to produce different outputs.

Lesson Outline

Introduction to Factor App (5 minutes)

1. Ask students: "What are the functions of the Factor App?"(Show factors of a number and check if it is prime or composite).
2. Ask students to fill in the blanks on Page 1 of *Student Guide: Lesson 3*.
3. Go over the answers, and review the flow and logic of the Factor App.

A. **Prime** Number?	B. Show all factors for the number.
C. Click on **Composite Number** button.	D. **Find factors** for the number.



Coding of the app (25 minutes)

1. Review the algorithm from Lesson 2 to find factors for a number.

   Count up from 1 to the Number:

   If remainder of (Number ÷ Divisor) = 0, then Divisor is a factor of Number.

2. Explain to students they will code a procedure called "findFactors" to create a list of factors for a number. The procedure will use a for each number loop to count from 1 to the Number. It will use a conditional block and the remainder block to determine is the divisor is a factor of the number.

3. Ask students to follow *Student Guide: Lesson* 3 to code the procedure and test that it accurately determines factors for a number.

Wrap-up (5 minutes)

1. Review the procedure "findFactors".

2. Tell students that they will code the two buttons, PrimeButton and CompositeButton in the next lesson.

Appendix 4
L3U8.4-8.5 Teacher's Guide: Lesson 4

Teaching Objectives

1. Focus: Help students to continue to deepen their understanding of conditionals by using them to determine if a number is prime or composite.
2. Encourage students to test and debug their apps fully.

Lesson Outline

Review Prime and Composite Numbers (10 minutes)

1. Ask students:
   (1) "Can you tell if 7 is a prime or a composite number?" (Prime)
   (2) "Why is 7 prime?" (Factors: 1 and 7 itself.)
2. Explain that the Factor app will calculate the number of factors of a number to determine if a number is prime. It counts how many factors it found in "FindFactors", and if the only factors are 1 and the number itself (i.e., only 2 factors), it is prime.
3. On the other hand, if the number of factors of the input number is more than 2, it is composite.
4. Ask students what conditionals they think they should use for PrimeButton and CompositeButton Click events.

Coding Activity (20 minutes)

1. Ask students to follow *Student Guide: Lesson 4* to complete the app. They will code the PrimeButton.Click and CompositeButton.Click event blocks.
2. (Optional) Have students to try *Student Guide: Challenge* to account for users entering 0 in the app.

Wrap-up (5 minutes)

1. Review the Computational Thinking Concepts learnt in this unit.
2. Review the Computational Thinking Practices used in this unit.
3. Explain the post-lesson activities.

Post-Lesson Activities

1. (Required): Complete multiple choice questions and survey of learning attitudes.
2. (Optional): Add other features such as scoring and a clock timer.