Stuck in a Debugging Loop

Meisen Wang | Fall 2024

Field(s) of Interest: Computer Science

Brief Overview:
In this lesson, mentees will learn about the basics of how we communicate to computers, and how we can write programs that are executed by computers. Mentees should understand binary code as the most basic form of computer code and that algorithms run in a strict and step-by-step manner. Mentees should understand that different outcomes can be programmed by if/else conditionals, and advanced mentees will learn how we can further modify the program.

Agenda:
- Introduction (5 min)
- *Module 0: Name Game (5-10 min)*
- Module 1: Commit to the Bit (10-15 min)
- Module 2: If BEAM: Return Yay! (10 min)
    - Can skip if low on time, add teaching goals into Module 3 instead
- Module 3: Beep Boop, Let's Loop! (10-15 min)
- Conclusion (5 min)

Main Teaching Goals/Key Terms:
- ➔ Binary code
- ➔ if/else
- ➔ algorithm

Background for Mentors

Module 1

- Language
- Binary code
- Compiler

Commit to the Bit

Humans speak many different languages to communicate with each other. Although computers don't quite "speak" English, Spanish, or Mandarin etc., we are able to communicate with them using computer **languages**. And just like how there are a multitude of human languages, there are many computer languages that programmers use to write code, like Java, Python, C++, etc. These languages, like human languages, may look different but can be used to convey the same meaning.

```java
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Figure 1: HelloWorld code in Java

As you may see, the languages that are used to write code consists of many English words. Computers aren't able to understand these words directly, and a program called a **compiler** will "translate" this human-written code into a machine code that the computer is able to understand. **Binary code** is the most basic form of computer code, consisting of 1s and 0s, which can be electronically represented as "on"s and "off"s. A binary digit, or bit, is the smallest unit of data. The American Standard Code for Information Interchange (ASCII), uses a 7-bit binary code to represent text and other characters. Each letter or symbol is assigned a number from 0 to 127. This is the system we will be using in the module 1 activity.

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Figure 2: ASCII binary code table

Module 2

- if/else
- and/or
- debugging

If BEAM: Return Yay!

When writing code, you may run into a situation where you want different outcomes for different conditions. This is where if/else conditional statements come in handy. The **if/else** statement allows the programmer to specify that *if* something is true, then do action1, *else* do action2. Any number of *else if* statements can also be added between the if and else statements to add more conditions and actions.

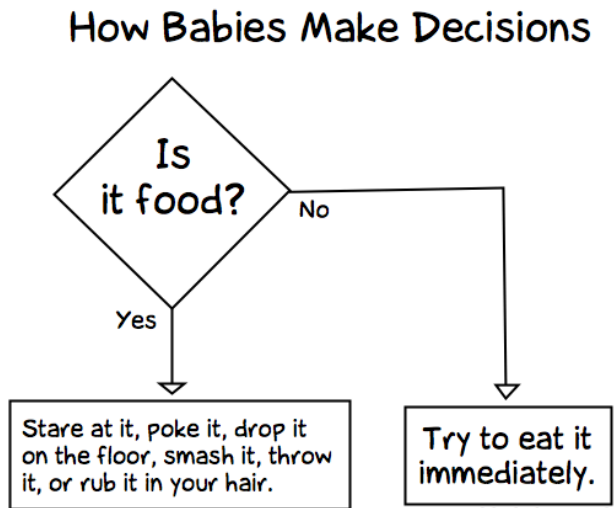


Figure 1: decision tree showing different outcomes for different conditions

The **and/or** operators allow conditions to be more specific or to include consideration of multiple conditions.

For example, the baby-decision-making diagram can instead say:
IF it's not food AND there is no parent nearby, then eat immediately.
In this case, both conditions need to be true for the action to occur.

An example using the or operator could be:
IF baby is hungry OR baby needs diaper changed, then cry.
In this case, only one condition needs to be true for the action to occur.

Debugging is the process of finding and fixing errors ("bugs") in code. These errors can range from simple typos to large-scale logical errors.

Module 3

- algorithm
- for/while loop

Beep boop, let's loop!

An algorithm is a process or set of rules to be followed in operation. This concept is present in many disciplines, but in the context of coding, an **algorithm** is step-by-step instructions that tell a computer how to solve a problem or do a task. Many complex tasks can be broken down into simpler steps, with alternate outcomes possible via if/else statements and repeated through loops.

Loops are used in programs in order to repeat a certain action multiple times, while a given condition is true. **For loops** and **while loops** will repeat the code inside of it until the condition is no longer true (meaning it can repeat forever if the condition is always true!). They have some technical differences, but can often be used interchangeably in simple programs.

Figure 1: Code inside of a loop is repeated until condition is no longer true

Introduction

Coding is increasingly relevant in various career fields and can seem daunting for students with little background, but the basic principles are quite simple! Mentees can see how computer programs can resemble our human logic in everyday life.

Concepts to Introduce	**Questions to Pique Interest**

• Computer:
  ○ Essentially “robots” that follow instructions
  ○ Able to handle large amounts of data and computation at a relatively fast speed
• Programmer:
  ○ The human giving the computer instructions!
	• Who here has used a computer? A website? AI?
	• How do you think a computer works? How does it know what to do?
	• What is coding?

Scientists, Current and Past Events	**Careers and Applications**
• The Turing Machine invented by Alan Turing in 1936 manipulated symbols on a strip of tape according to a table of rules. ○ Movie: The Imitation Game • ChatGPT: uses specialized algorithms to find patterns within data. It isn’t given exactly what the patterns are, but its algorithm allows it to make predictions while it’s trained to be optimized.	• Software Engineer • Data Scientist • UI/UX Design

Module 1: Commit to the Bit

This is an individual activity where mentees will string beads into a bracelet/lanyard to visualize binary code.

Teaching Goals

1. **Language:** what's used to communicate with a computer
2. **Binary code:** the most basic form of computer code, consisting of 1s and 0s
    a. The "language" that computers "speak", made of 1s and 0s, or black and whites
3. **Compiler:** a program that converts typed code into a code that can be read and executed by a computer

Materials

Per student:

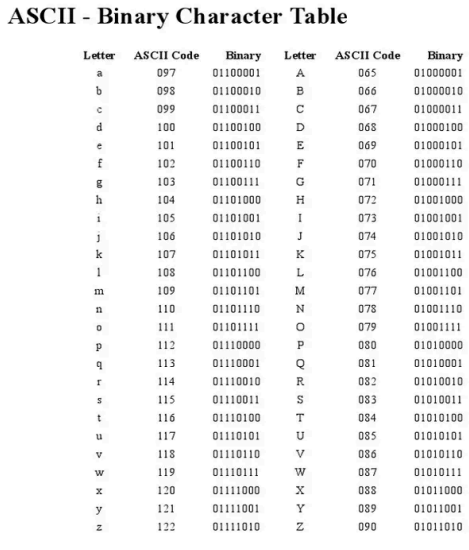
- Pipe cleaner
- Beads (2 colors)
- ASCII binary code printout

Different Methods for Teaching

- **Communication:** A great way to get mentees more introduced to the idea of computer programming languages is to remind them that people can communicate in different ways. Compare programming languages to spoken languages and talk about how both are forms of communication.
- **Breaking down information:** Discuss examples of ways in which language and information is broken down to be easier to understand. Relate letters in the alphabet to the 1s and 0s of binary code as both are the most basic form of each language.
- For older mentees, you can relate the concept of compilers to the general concept of dictionaries. A dictionary is a book that helps people understand what words mean in the same way a compiler is a system that helps computers be able to read code.

Procedure

1. String beads onto pipe cleaner in order, according to the ASCII binary printout, for mentees' first and last initial.
    a. 1 => white bead, 0 => navy bead (i.e. MW => 01001101 01010111)
    b. Separate first and last initial with one bead of a random color (not white or navy)
2. Bracelet option vs lanyard option:
    a. Bracelet: Place "bracelet" around the wrist and twist together the ends of the pipe cleaner to secure.
    b. Lanyard: tie a knot on both ends or tie



ASCII - Binary Character Table

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

one end on a backpack!	**Figure 1:** ASCII binary code table.  **Figure 2:** MW => 01001101 01010111  **Figure 3:** bracelet option  **Figure 4:** lanyard option

Classroom Notes

Younger mentees can use just their first initial instead.

Module 2: If BEAM: Return Yay!

This is a class activity where mentees will act out how a program runs systematically.

Teaching Goals	**Materials**
4. **If-else:** "if this condition is true do the following thing, else do this thing instead" 5. **And, or:** operators that modify the conditional so that multiple conditions must be true or one condition must be true 6. **Debugging**: the process of finding and fixing errors in code	• notecards

Engaging the Mentees

Sometimes, it can be difficult to engage the mentees in more interactive activities, especially if it's the end of the day. Some tips:

- Don't pressure the mentees into participating if they refuse to! Instead, demonstrate the activity with the mentees/mentors who do show interest, and this will ideally allow the mentees to feel more ready to contribute
- Ask the mentees 1 on 1 if they want to participate and specifically assign them a role, so that they feel more seen during the lesson.

Procedure

Option 1 (for younger sites): Follow the Leader(s)

1. Mentors & mentees agree on a set of rules, for example:
   - if Mentor1 is touching his nose:
     - ➔ mentees also touch their nose
   - if Mentor2 rubs their belly:
     - ➔ mentees also rub their belly
   - if Mentor3 AND Mentor4 point up:
     - ➔ mentees jump up and down
   - if Mentor3 OR Mentor4 squat down low:
     - ➔ mentees squat down low
2. Play the game! Mentees who mess up are "out"

Option 2 (for older sites)::

1. Pass out one notecard to each mentee

Mentees all have notecards with their own if/else

Figure 1: (Example) Mentees raise their hands if Mentor performed a certain action

instructions, for example: If you get tapped on the head: If there is someone wearing a blue shirt in this room: ➔ Go and tap them on the head. Then, return to your seat. else: ➔ Stand up and yell “No one is wearing a blue shirt!” Else: ➔ stay sitting down 2. Begin by tapping 1 mentee on the head. 3. This process may run on forever as 2 mentees may tap each other back and forth. How can we fix this? (debugging)	 **Figure 2:** *Boy in green shirt is tapped on the head*

Classroom Notes

Module 3: Beep Boop, let's loop!

This is a group activity where mentees can use their creativity to design an algorithm that breaks down a certain complicated action into small steps.

Teaching Goals	**Materials**
7. **Algorithm:** a process or set of rules to be followed in operation a. step-by-step instructions that tells a computer (or a person) how to solve a problem or do a task 8. **(for, while) Loop:** running the same block of code over and over again, with certain rules	• none

Adapting activities for your classroom

Every class is different, and with an activity so open-ended, there are many ways to adapt it to fit the interests of your class.

- If the teacher allows, try doing this activity outside to give the mentees more room for inspiration
- If mentees are younger, and will likely take lots of time to write instructions, have the mentees raise their hands and call out instructions one at a time instead.

Procedure

1. Split the class into as many groups as mentors.
2. Each group of mentees write a "recipe"/step-wise instructions (algorithm) for a mentor "robot" to perform a certain action
   c. Options: making a PB+J sandwich, jumping jacks, macarena, walk around the room with no collisions, choose your own!
   d. Mentees should use if/else, and/or, and loops in their algorithm to make their instructions more specific and foolproof
3. Mentors act out the task described by their mentees as a "robot", taking instructions as literally as possible

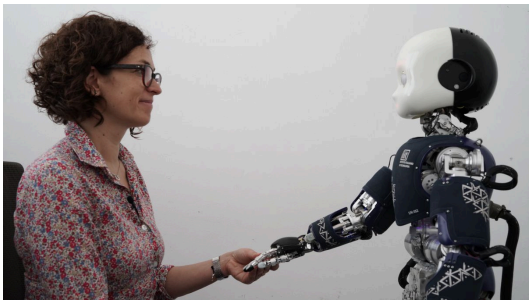


Figure 1: *Mentee (left) and Mentor (right)*

Classroom Notes

Younger mentees should probably pick from the options given, while older mentees can try to design their own algorithm for a more complex action/task.

Conclusion

Mentees should see that coding may not be as daunting as they had imagined!

References

- Add references in case your mentors want additional information!
- Title of Source, Author, Organization. http://www.example.com/

Summary Materials Table

Material	**Amount per Site**	**Expected $$**	**Vendor (or online link)**
Beads	17 per student		
Pipe cleaners	1 per student		