ENR145 Computational Methods: “Hamming Sheets” recap and “Hamming Python” 101

Xiang Li

Spring 2026

COE COLLEGE

Hamming Sheets

- Hope everyone learned something new with either Hamming or spreadsheet.
- I will strive to make my instruction more clear. **Reading comprehension is NOT the learning objectives in this class.**
- I learned something new from you guys as well.

Examples of hamming sheets done by the students:

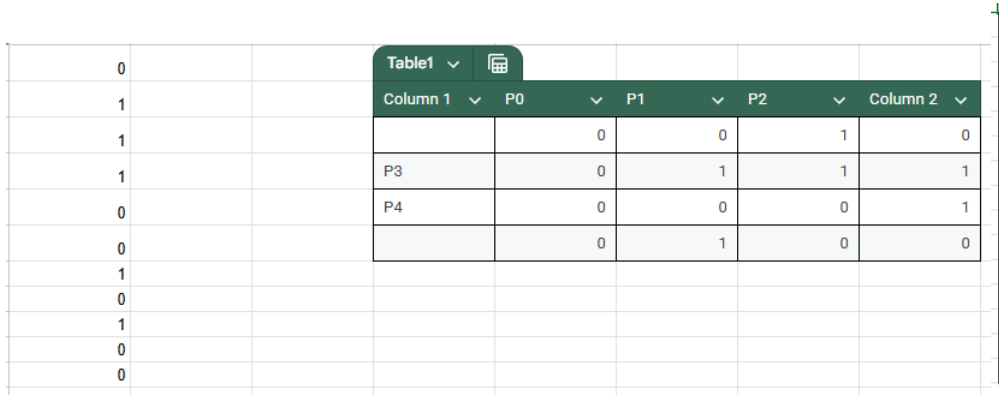
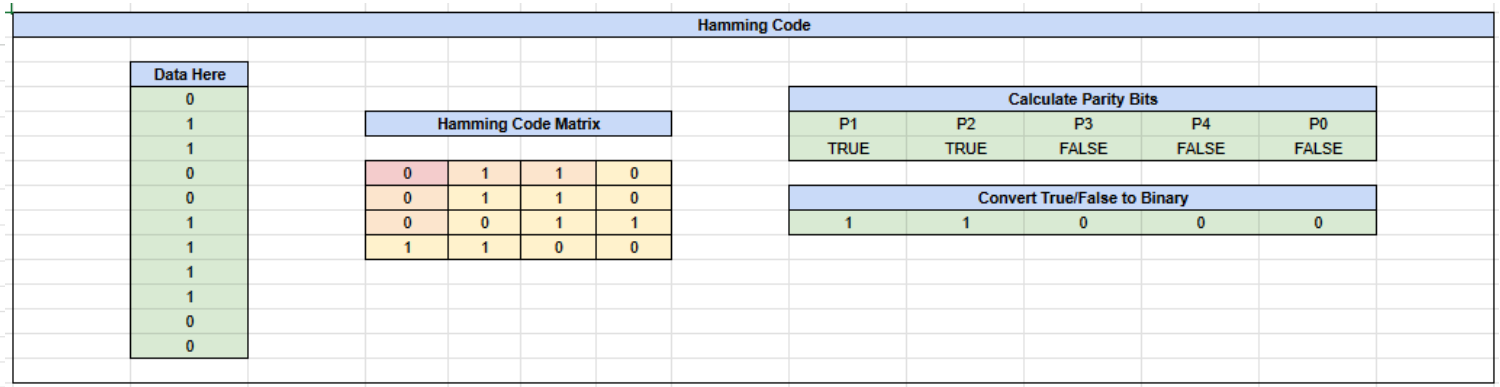


	Table1				
	Column 1	P0	P1	P2	Column 2
0		0	0	1	0
1	P3	0	1	1	1
1	P4	0	0	0	1
1		0	1	0	0
0					
0					
1					
0					
1					
0					
0					



Hamming Code

Data Here
0
1
1
0
0
1
1
1
1
0
0

Hamming Code Matrix

0	1	1	0
0	1	1	0
0	0	1	1
1	1	0	0

Calculate Parity Bits

P1	P2	P3	P4	P0
TRUE	TRUE	FALSE	FALSE	FALSE

Convert True/False to Binary

1	1	0	0	0

COE COLLEGE

Revisit Error Checking in Hamming Sheets

Hamming Code Matrix			
0	1	1	0
0	1	1	0
0	0	1	1
1	1	0	0

Hamming Code Matrix Recieved			
0	1	1	0
0	1	1	0
0	0	1	1
1	1	0	0

Error Code				
0	0	0	0	0

No bit flipped

Hamming Code Matrix Recieved			
0	1	1	0
0	1	1	0
0	0	1	1
1	0	0	0

Error Code				
1	1	0	1	1

1 bit flipped

Hamming Code Matrix Recieved			
0	1	1	0
0	1	1	0
0	0	1	1
1	0	1	0

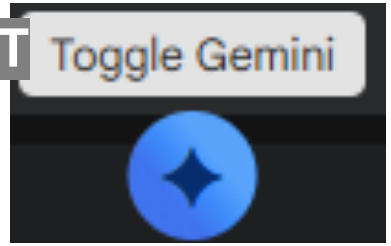
Error Code				
0	0	1	1	0

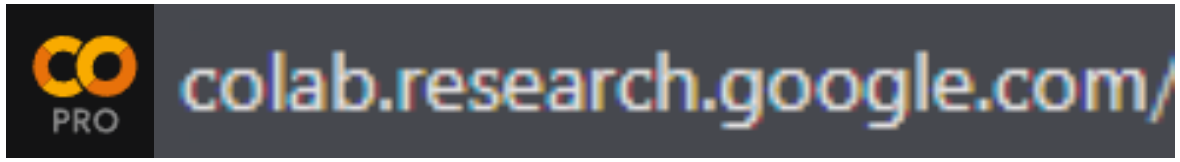
2 bits flipped

COE COLLEGE

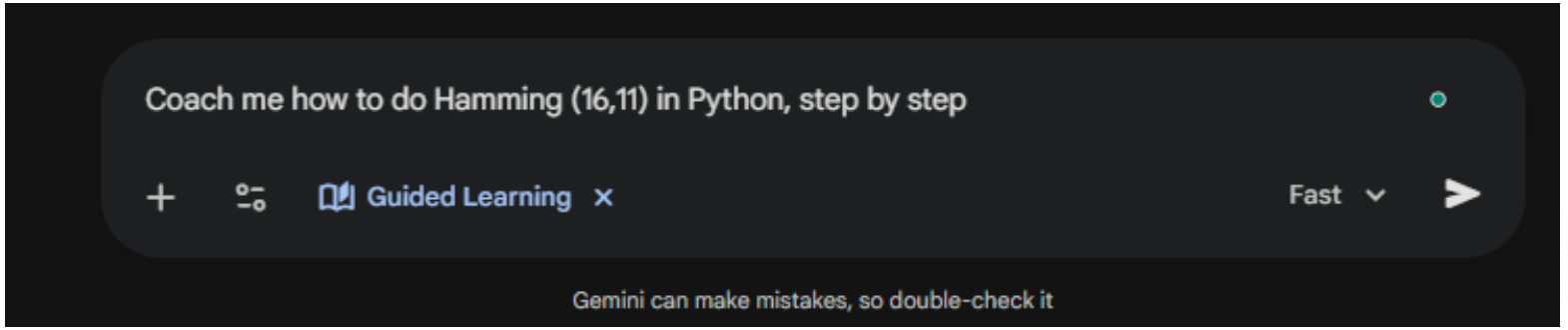
Before we move into the land of "Hamming Python", here's the deal:

When in class:

NOT 

in 

When off class:



COLLEGE

Let’s go, hamming python 101

ENR 145: Computational Methods for Physicists and Engineers

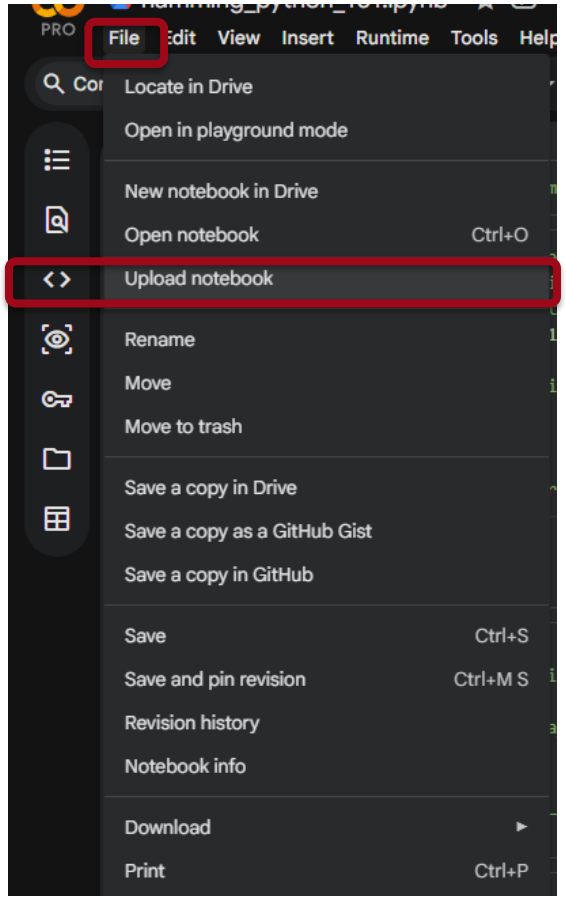
Department of Engieering Physics, Coe College | Cedar Rapids, Iowa

Download Syllabus | Upload to Moodle

Module 1: Codes, Visuals, and Algos (4 weeks)

Week 2: Google Sheets helper file

Hamming python 101(right click and "save link as" to download)



COE COLLEGE

What if I already KNEW how to do this?

- You are free to go.

- To get extra class token:

1. Work out the codes in a Python IDE (VS Code, PyCharm, etc… up to 1 token)
2. Work out the codes in a new language (C, C++, Rust, html, etc… no token cap)

- Starting Python session in 3, 2, 1…

From 0 to (16, 11) in 4 steps

ENR145 modular 1: let's do hamming extended encoding without any external python library.

› Step 1: We need a data format to store 11 bits and 16 bits of data, how?

↳ 6 cells hidden

› Step 2: What wiil be your solution to do boolean operation with number systems?

↳ 5 cells hidden

› Step 3: Let's visit/revisit conditions and loops:

↳ 7 cells hidden

› Step 4: Let's do hamming (16,11)

↳ 9 cells hidden

+ Code + Text

COE COLLEGE

From 0 to (16, 11) in 4 steps: step 1

Discuss:

- **For all the data types, why not use number to store numbers?**

```
input_data_1 = 11000100010 # decimal number (or binary number)?
input_data_2 = "11000100010" # string
input_data_3 = [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0] # a number list aka array

print (type(input_data_1)) # this is the way to check data type
print (type(input_data_2))
print (type(input_data_3))
# print (len(input_data_1))
#print (len(input_data_2)) # this is the way to check data lenght when applicable
#print (len(input_data_3))
```

```
<class 'int'>
<class 'str'>
<class 'list'>
```

COE COLLEGE

From 0 to (16, 11) in 4 steps: step 1

Discuss:

- **How to manipulate a single bit in that data stream/number array?**

From 0 to (16, 11) in 4 steps: step 2

From pre-labs, you are tasked to look up Booleans in Python (AND, OR, NOT, XOR)

- **Now let's figure out how to bool numbers.**

From 0 to (16, 11) in 4 steps: step 3

From pre-labs, you are tasked to code Karel the Robot, where it has all the conditions (if, while) and loops (for) you can practice.

- **Let's try out the IF and For, and nested For in python**

From 0 to (16, 11) in 4 steps: step 4

Remaining road blocks for hamming

- **Look-up table**
- **Assign value based on index**
- **Assign parity check**