Midterm Review

Review mock, go over your questions

Emily Zhao
CSCI-UA-002

Exam Info

- In-person, in classroom
- Time limit: 1 hour 15 minutes
- Paper exam
  - Will be scanned, writing in pen recommended
  - Scratch paper will be provided!

Question Types

- Short/long fill-in-the-blank
  - What's the output?
  - If it's an error, what type?
- Reordering code
- Debugging and finding code errors
- **Long programming questions**
  - These are worth the most points so I recommend starting with these!
  - I will also give partial credit for pseudo code that has good logic!

Python Core Language Elements & Functions

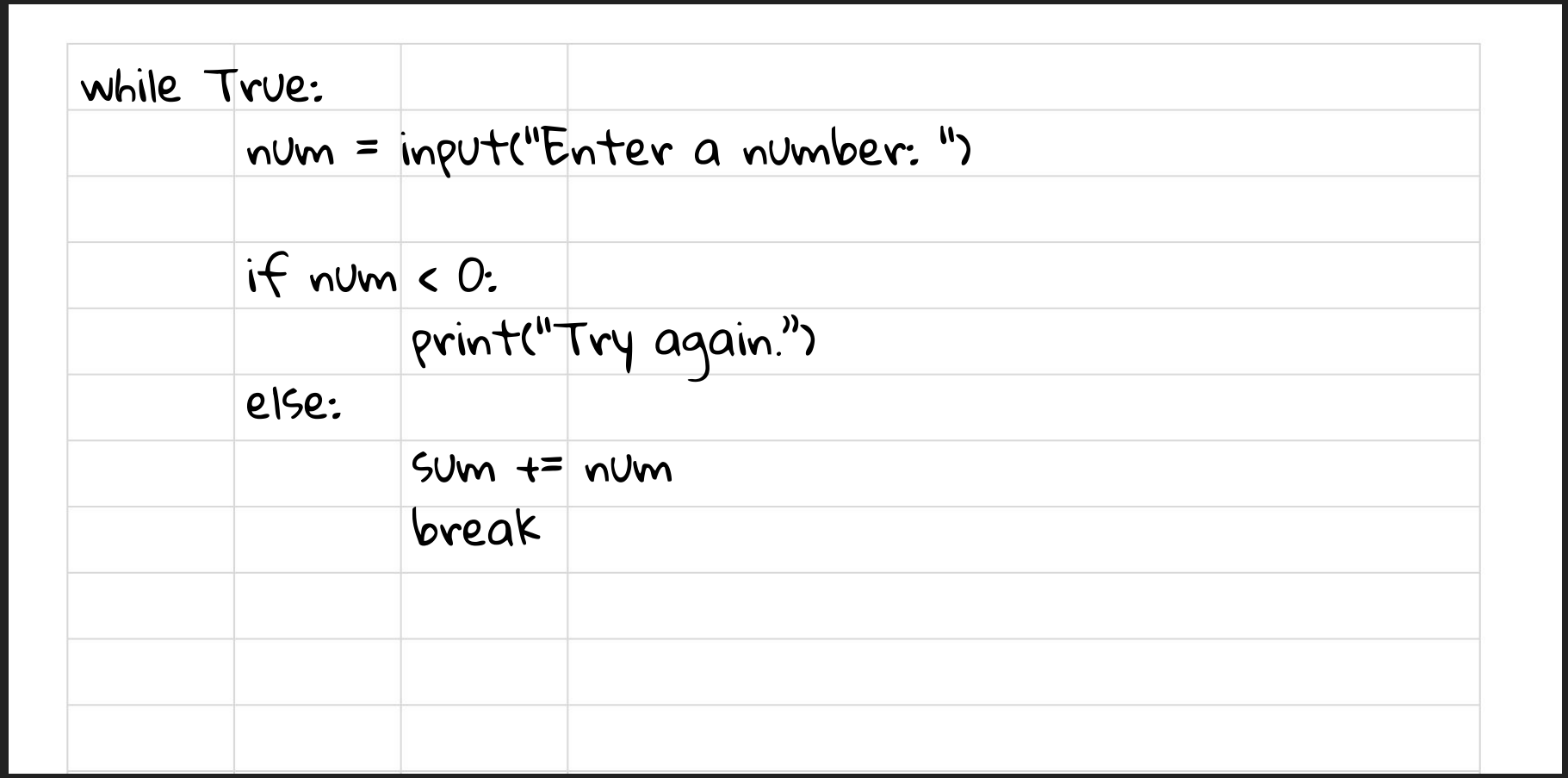
```
and             input           Module Functions
def             int             random.randint
elif            len             str.lower
else            not             str.upper
float           or              time.time
for             print
format          return
if              str
in              while
import
```

ASCII Code Table

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

Decimal ASCII Chart

Long programming paper



^ ^ ^ ^ Lines are guides for indentation

Topics Covered

Module 1: Variables, Statements, Etc...
Module 2: Types, Operators, Debugging
Module 3: Boolean Logic / Conditionals
Module 4: While Loops
Module 5: For Loops, Nested Loops
Module 6: Functions

More details can be found in previous class slides:
module-06-midterm-review

Starred Topics

- Modulo
- Escape Characters
- Error Types
- Format
- Functions

Modulo (%)

- The **remainder** operator
- If you divide the second number by the first number, what is the remainder?

  12 % 5 → 7
  60 % 30 → 0
  4 % 7 → 4

- Good for isolating numbers places

Escape Characters

- Any text you put between two matching delimiters (", ' , '''') will become a string!
  - A named variable loses its variable status as soon as you put it between delimiters, as well. (i.e. "<widths" is becomes a string and will not be replaced with the value of width)
- The backslash character (\ ) acts as a signal to denote the following characters as special (aka “do not stringify anything that follows me”)
- Examples: \t, \n, \"

Error Types

Syntax: code won’t even save / grammatical

- Are you missing a comma/parentheses?
- Do your delimiters match?
- Did you misspell print?

Runtime: happens while the program is running

- User didn’t enter in the data type you were expecting
- Division by zero

Logic: you wrote code that doesn’t get you your desired results, will run fine, and will also save

Format

- Returns a **string**
- Helps you control the **TOTAL** width of a string, thus useful for making tables
- Usually recommended for use in a final print statement/at the end of your program when you need to display

Functions

Input, Processing, Output/Return

- Functions can take in optional inputs, they all do something, and they can optionally return something, as well
- Value-returning functions can be stored in a variable or printed

print( )

Input: optional

Do something: If an input is received, write the input on the blackboard; f no input is received, write a new line character on the blackboard

Output: none

random.randint( )

Input: two integers

Do something: calculate a random integer between the two inputs

Output: random integer calculated

```
def mashify(name):
    output = ""
    for letter in name:
        output += letter + " * "
    return output

mashify("Emily")
```

```
def mashify(name):
    output = ""
    for letter in name:
        output += letter + " * "
    return output

mashify("Emily")
```

Since this function returns something, I need to either store it in a variable or print it!

```
def mashify(name):
    output = ""
    for letter in name:
        output += letter + " * "
    return output

mashify("Emily")
```

Since this function returns something, I need to either store it in a variable or print it!

```
def mashify(name):
    output = ""
    for letter in name:
        output += letter + " * "
    return output

print(mashify("Emily"))
```

Homework

Study for the midterm! Good luck!

Assignment #6 also due