

CS 1110/1111 Exam 1

Name _____

Bubble in your computing ID in the footer of this page. We use an optical scanner to read it, so fill in the bubbles darkly. If you have a shorter ID, leave some rows blank. In case we have trouble with the scanner, please also legibly print your computing id at the top of each page. Also print your name on this page.

Please follow these rules throughout the exam:

- Write legibly; we can only give full credit for answers we can recognize as correct.
- Indentation and punctuation do matter.
- Write on the lines, where possible. The lines are numbered so we can reference them when giving grading feedback.
- If you need to insert a line between two you've written, make it clear what you are doing.
- We grade one page at a time. Do not spill answers onto another page.
- Don't add features we didn't request: only print if we ask you to print, etc.
- Don't include docstrings for your functions (unless a question says otherwise).

The exam is being given in multiple locations simultaneously, so we cannot fairly answer student questions during the exam. **If you find a question ambiguous or unclear**, write that down on your exam next to the question and we will give it due consideration during grading.

1. [1 points] Pledge

On my honor as a student I have neither given nor received aid on this exam.

signed: _____

2. [1 points] Computing id

Put your computing id on the top of each page.

3. [2 points] Bubbles

Fill in the bubble footer with your computing id.

(the bubble footer is automatically inserted into this space)

Page 2:

4. [10 points] Assigning

Complete the following table (there are 10 cells for you to fill). The first row is done for you.

Value of a	Code that was run	New value of x	Type of x
5	<code>x = a * 3</code>	15	int
2.5	<code>x = str(a) * 2</code>	2.52.5	str
10	<code>x = a / 2</code>	5.0	float
5	<code>x = (a == 5)</code>	True	bool
12	<code>x = a % 4</code>	0	int
7	<code>x = a // 2</code>	3	int

1 point per cell

Page 3:

5. [11 points] Displaying and Evaluating

Fill in the following table.

- “Displayed in Run Window” refers to anything that shows up in the output window, whether printed, a prompt, or error message.
- If the code does not display anything then write “Nothing”.
- If it’s an error message, answer “Error” in either column (leaving the other column blank).

The first row is done for you

Code	Displayed in Run Window	Evaluates To
<code>2 + 3</code>	Nothing	5
<code>3 ** 2</code>	Nothing	9
<code>'bye' * '3'</code>	Error	
<code>print(float(12))</code>	12.0	None
<code>9 == 9</code>	Nothing	True
<code>float(int(3.2 + 0.9))</code>	Nothing	4.0
<code>print(str(4 + 7))</code>	11	None
<code>print(3 >= 4)</code>	False	None

1 point per cell. Treat the two cells on the row with the error as one cell, error can be written in either cell

Page 4:

The following program runs without generating any errors, but does not give the output that the programmer was expecting.

```
1 def withdraw(amount):
2     global balance
3     balance -= amount
4 def deposit(amount):
5     global balance
6     current_balance = balance
7     current_balance += amount
8 def bank_statement():
9     global balance
10    if balance >= 0:
11        print("Balance:", balance)
12    else:
13        print("Overdraft:", balance)
14
15 balance = 1000
16 bank_statement()  # What is printed? _____
17 withdraw(800)
18 bank_statement()  # What is printed? _____
19 deposit(500)
20 bank_statement()  # What is printed? _____
21 withdraw(300)
22 bank_statement()  # What is printed? _____
```

6. [4 points] Statement balance

Trace through this program and in the four blanks above (lines 16, 18, 20, and 22), give the output (What is printed?) each time the `bank_statement()` function is called.

Balance: 1000 Balance: 200 Balance: 200 Overdraft: -100

7. [3 points] Global and local variables

For each of the 3 variables given below, determine if the variable is a local variable or a global variable (circle the word local or the word global at the end of each statement).

The variable `balance` on line 15 is - local global **GLOBAL**

The variable `balance` on line 3 is - local global **GLOBAL**

The variable `amount` on line 3 a is - local global **LOCAL**

1 point per item on this page

Page 5:

8. [10 points] Print some letters

Use the function `letter()` shown below to determine what will be printed each time the function is called and passed a certain argument. For each row of the table, trace the function call in the first column and put an x in the column of each letter that will be printed. Some rows may have multiple checks. If nothing is printed, check the column labeled “nothing”. The first row has been completed for you.

```
def letter(num):
    if num == 10:
        print('a')
        return

    if num < 10:
        print('b')
        if num < 5:
            print('c')
    elif num < 20:
        print('d')
    elif num < 30:
        print('e')
        return
    else:
        print('f')

    if num < 40:
        print('g')
```

Function Call	a	b	c	d	e	f	g	nothing
letter(10)	X							
letter(3)								
letter(7)								
letter(25)								
letter(35)								
letter(100)								

letter(3) b c g
 letter(7) b g
 letter(25) e
 letter(35) f g
 letter(100) f

2 points per row

1 point for not having more than one incorrect cell in the row

1 point for not having any incorrect cells in the row

Page 6:

9. [8 points] Hospital Charity Banquet

You are working with a team of volunteers to organize a hospital charity banquet. You are trying to invite employees and need to write a function to help you format the names. Most of the employees have a preferred title indicated (such as 'Mr', 'Ms', 'Nurse', 'Dr', etc.). But for some people, the preferred title is unknown. You need to write a function named `invite` that will accept two `str` parameters, a name and a title. But the function must also be able to accept just one `str` parameter, a name. If someone's title is unknown, just assume that it is 'Dr'. The function must return a single string containing the appropriate greeting. For example if we ran:

```
print(invite('Lee', 'Ms'))
print(invite('Reynolds', 'Mr'))
print(invite('Strange'))
```

Your program should print out this exactly:

```
Ms Lee
Mr Reynolds
Dr Strange
```

[SAMPLE SOLUTION]

```
def invite (name, title="Dr"):
    return(title + ' ' + name)
```

[Rubrics?]

- 1.) there is a function header that includes `def` and `invite`
- 2.) function has 2 parameters
- 3.) exactly 1 parameter has an `=`
- 4.) the second parameter is correctly assigned 'Dr'
- 5.) a name and a title are put together
- 6.) a space is put between the title and the name
- 7.) a value is returned
- 8.) the function is completely correct

Page 7:

10. [10 points] Triple Cool

Your math professor loves strange numbers. They have invented a new category of number that they call “Triple Cool” (TC) numbers. A TC number is a 6-digit number that looks like a 2-digit number that repeats 3 times, such as: 414141 or 272727. To be a TC number, the number must also be odd. Your professor wants you to write a program that asks the user to enter a 2-digit integer on the keyboard and tests to see if that number could make a TC number. The example below shows the program run three separate times, entering the numbers 41, 12, and 99 respectively:

```
Enter a number: 41
414141 is TC!
```

```
Enter a number: 12
121212 is not TC
```

```
Enter a number: 99
999999 is TC!
```

[SAMPLE SOLUTION]

```
two_chars = input('Enter a number: ')
six_chars = two_chars + two_chars + two_chars
num = int(six_chars)
if num % 2 == 0:
    print(num, 'is not TC')
else:
    print(num, 'is TC!')
```

[Rubrics?]

- 1.) an input function is used
- 2.) the value returned from the input function is stored in a variable, or used in a meaningful way
- 3.) a two-character string is used to produce a 6 character string
- 4.) a number is cast to an int type before checking for odd
- 5.) checking for odd uses an appropriate comparison
- 6.) an if statement exists that has a boolean expression to evaluate
- 7.) the if statement produces two possible outcomes
- 8.) the logic of the if statement is correct
- 9.) the program includes a way to produce two different messages
- 10.) the program is totally correct

Page 8:

11. [10 points] Mars Exploration

You have written a program to manage a small exploration mission on the surface of Mars. When rovers are sent out from base camp, they can go either east or west. You need to add a function to your program called `direction` that will return either the string "east" or the string "west". The first value returned should be "east", then the values should alternate each time `direction` is called. For example, if `direction` was called four times in a row:

```
print(direction())
print(direction())
print(direction())
print(direction())
```

The program would print out:

```
east
west
east
west
```

[SAMPLE SOLUTION]

```
x = 'east'
def direction():
    global x
    if x == 'east':
        x = 'west'
    else:
        x = 'east'
    return x
```

[Rubrics?]

- 1.) a function named `direction` is defined
- 2.) the function has no parameters
- 3.) a global variable exists outside of the function
- 4.) the global variable is used inside the function with the keyword `global`
- 5.) the state of the global variable is checked inside the function
- 6.) there is correct logic for toggling between two values
- 7.) the if statement produces two possible outcomes
- 8.) the logic of the if statement is correct
- 9.) the correct value is returned
- 10.) the program is totally correct