

CS 301 - Spring 2018
Instructor: Laura Hobbes LeGault

Midterm Exam 1 — 16.67%

(Last) Surname: _____ (First) Given name: _____

NetID (email): _____ @wisc.edu

IMPORTANT: Answers for all questions *must* be marked on a scantron. The answer marked on the scantron will be the only answer graded.

Fill in these fields (left to right) on the scantron form (use #2 pencil):

1. LAST NAME (surname) and FIRST NAME (given name), fill in bubbles
2. IDENTIFICATION NUMBER is your Campus ID number, fill in bubbles
3. Under *ABC* of SPECIAL CODES, write your **lecture number** and fill in:
 001 - MWF 8:50a (Hobbes morning)
 002 - MWF 1:20p (Hobbes afternoon)
 003 - TR 1:00p (Paul Barford)
4. Under F of SPECIAL CODES, write A (exam version), fill in bubble 0

.....

I certify that I will keep my answers covered and do my best to not allow my exam paper to be viewed by another student during the exam or prior to completion of their exam. I also certify that I have not viewed or in any way used another's work in completing my answers. I understand that being caught allowing another to view my work or being caught viewing another's work are both violations of this agreement and either will result in automatic failure of the course and an academic misconduct letter to the Deans Office for myself and any other individuals involved.

Signature: _____

.....


The following exam has 25 questions and is worth a total of 42 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 12 questions worth 1 point each. Choose the *best* answer.
2. **Multiple Choice** — 9 questions worth 2 points each. Choose the *best* answer.
3. **Code Completion** — 4 questions worth 3 points each. Be complete.

You may not use notes or books, your neighbors, or calculators or any other electronic devices on this exam. **Turn off and put away** any portable electronics now.

Disclaimer: the following are provided for your reference only, and the inclusion of information here does not guarantee it will be used on the exam.

Operator Precedence Table:

level	operator	description
higher  lower	(<expression>)	grouping with parentheses
	x[index]	indexing
	* / %	multiplicative
	+ -	additive
	< <= > >=	relational
	== !=	equality
	not	logical not
lower	and	logical and
	or	logical or
	= += *=	(compound) assignment

Built-in functions:

`raw_input(p)` Prompts the user for input using `p` and returns the input as a string.
`len(s)` Return the length (the number of items) of an object.
`type(x)` Returns the *data type* of the value stored in `x`.
`int(x)` Returns the integer representation of `x`. `ValueError` if not possible.
`float(x)` Returns the float representation of `x`. `ValueError` if not possible.
`str(x)` Returns the string representation of `x`.

Constants and functions from the math module:

`math.sqrt(x)` Returns the square root of `x` as a `float`.
`math.pow(x,y)` Returns `x` raised to the power `y`. Converts both arguments to `floats`.
`math.pi` The mathematical constant $\pi = 3.1415\dots$

String methods:

`str.index(s)` Return the lowest index in the string where substring `s` is found.
`str.isalpha()` Return `True` if all characters in the string are alphabetic.
`str.isdigit()` Return `True` if all characters in the string are digits.

Dual Choice: Terminology

1. The **value** provided to a function when the function is **called** is the _____. (1)
 - A. argument
 - B. parameter
2. **Changing** a value from one type to another is called _____. (1)
 - A. type conversion
 - B. type concatenation
3. The operation to determine **remainder** after integer division is called _____. (1)
 - A. truncation
 - B. modulo
4. A function's **name** _____ begin with a digit (number). (1)
 - A. cannot
 - B. can
5. When calculating a value to **use** in another function, you should _____ the value. (1)
 - A. return
 - B. print
6. A **local** variable is defined _____ a function. (1)
 - A. inside
 - B. outside
7. In the expression `5 % 2`, the component 5 is called an _____. (1)
 - A. operand
 - B. operator
8. With **mutually exclusive** conditions, a minimum of _____ **must** evaluate to True. (1)
 - A. zero (0) conditions
 - B. one (1) condition

True or False: Evaluating boolean expressions

9. True or (True and False) (1)
A. True
B. False
10. `len("string") != 37 % 8` (1)
A. True
B. False
11. `"9.0" == "9"` (1)
A. True
B. False
12. `1/2 < 0.5` (1)
A. True
B. False

Multiple Choice: Reading code

13. What is the **data type** of `x` after the following line of code is executed? (2)
`x = raw_input("How many points did you get on the exam?") / 42`
A. This code would produce an error.
B. float
C. int (integer)
D. str (string)
14. Which of the following function calls will **always** cause `can_move_up(up, left, right)` implemented below, to return **True**? (2)

```
def can_move_up(up, left, right):  
    if type(up) == int:  
        return True  
    return False
```

- A. `can_move_up(up, left, right)`
B. `can_move_up(5, None, None)`
C. `can_move_up(1.0, 2, 3)`
D. `can_move_up(5)`

15. What is the **value** in `x` after the following line of code is executed? (2)

```
x = "hello world".index("w") + 1
```

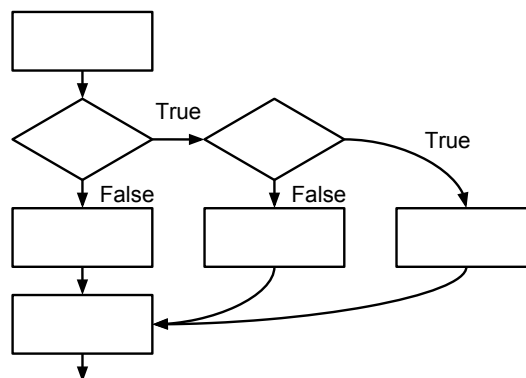
- A. "w1"
- B. 7
- C. 8
- D. This code would produce an error, cannot use + with str and int objects.

16. What is the **value** in `x` after the following code executes? (2)

```
x = 0
while x > 10:
    x -= 3
```

- A. -12
- B. -3
- C. 0, the loop never runs
- D. The loop is infinite.

17. Which of the following best describes the control structure pictured in this flow chart? (2)



- A. if/else
- B. if/else, where the if contains an if/else
- C. if/elif/else
- D. if/elif

18. What is the *error* produced when the following code is executed? (2)

```
x = "hello world"(11.0)
```

- A. IndexError: string index out of range
- B. TypeError: string indices must be integers, not float
- C. TypeError: str object is not callable
- D. This code does not cause an error.

19. What is the **data type** of **x** after the following line of code is executed? (2)

```
x = (15 % 2) * 6 > 145 / 20
```

- A. float
- B. bool (boolean)
- C. int (integer)
- D. str (string)

20. If **x** initially contains the value 0, what value does it contain after it is **decremented** ten (10) times? (2)

- A. 10
- B. -10
- C. 0
- D. Decrementing a value of this type would produce an error.

21. What is the value in **x** after the following code executes? (2)

```
def detect_even(num):  
    if num % 2 == 0:  
        return True  
  
x = detect_even(5)
```

- A. False
- B. True
- C. None, the function does not return a value.
- D. 1

Code Completion: Writing code

For each of the following questions, select the value, operator, or statement needed to produce the indicated output (check the comments if you need a hint).

22. Select the correct condition to complete this implementation of this `is_heronian()` function, assuming the `area()` function exists and returns the area of the triangle. Recall that a triangle is Heronian if its edges and area are whole numbers. (3)

```
def is_heronian(a, b, c):  
    tri_area = area(a, b, c)  
    if _____ :  
        return True  
    return False
```

A. `type(tri_area) == int`
B. `a%1 == b%1 == c%1 == 0`
C. `int(tri_area) == tri_area`
D. B and C (and is an operator)
E. A and B (and is an operator)

23. Select the correct condition for the following counter controlled loop so that it will print the word **hi** **three (3)** times. (3)

```
counter = 0  
while _____ :  
    counter += 2    # careful!  
    print "hi"
```

A. `5 != counter`
B. `5 > counter`
C. `counter != 3`
D. `counter < 3`

Fill in the following **two** blanks to find the end of a sequence of consecutive characters in the string `data` and add the letter and sequence length to the string `compress`.

24. `start = # some index`
`stop = start + 1`
`while _____ :`
 `stop += 1`
`num_consecutive = stop - start`
A. `stop < len(data)` (3)
B. `data[stop] == data[start]`
C. `data[stop] != data[start]`
D. A and B (and is an operator)
E. A and C (and is an operator)
25. `compress += data[start]`
`if stop - start > 1 :`
 `compress += _____`
A. `"stop-start"` (3)
B. `data[start] * (stop-start)`
C. `str(stop-start)`