CS 301 - Fall 2016

Instructor: Laura Hobbes LeGault

Midterm Exam 1 — 16.67%

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

NetID (email): _____ @wisc.edu

IMPORTANT: Answers for Dual and Multiple Choice 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 001 (morning lecture), fill in bubbles
4. Under J 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 26 questions and is worth a total of 86 points. You will have 50 minutes to complete the exam. **Be sure to read through every question completely.**

1. **Dual Choice** — 13 questions worth 2 points each. Choose the *best* answer.
2. **Multiple Choice** — 9 questions worth 4 points each. Choose the *best* answer.
3. **Fill-in-the-blank** — 4 blanks worth 6 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
	`( <expression> )`	grouping with parentheses
higher	`x[index]`	indexing
	`* / %`	multiplicative
	`+ -`	additive
↕	`< <= > >=`	relational
↕	`== !=`	equality
	`not`	logical not
	`and`	logical and
lower	`or`	logical or
	`= += *=`	(compound) assignment

Built-in functions:

`input(p)`	Prompts the user for input using `p` and returns the user's input.
`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 `float`s.
`math.pi`	The mathematical constant $\pi = 3.1415...$

String functions:

`w.isalpha()`	Return true if all characters in `w` are letters and `w` is not empty.
`w.startswith(z)`	Return true if `w` begins with `z`, otherwise return false.

Functions from the random module:

`random.randint(a,b)`	Return a random integer N such that `a <= N <= b`.

Dual Choice: Terminology

1. *Casting* a variable changes its _____. (2)

   A. type
   B. value

2. To *assign* a value to a variable, you must use the operator _____. (2)

   A. `=` (one equals sign)
   B. `==` (two equals signs)

3. You see some code that includes the line (2)

   ```
   print x()
   ```

   Based on what you know about Python, `x` must be a _____.

   A. function
   B. variable

4. A _____ is an example of an *output* device. (2)

   A. mouse
   B. printer

5. The following line of code is called the _____. (2)

   ```
   def mean(a, b, c, d):
   ```

   A. arguments
   B. function header

6. A function that does not include a return statement always returns the value _____. (2)

   A. `False`
   B. `None`

7. Long-term (persistent) memory is a responsibility of the _____. (2)

   A. hard disk drive
   B. RAM

8. *Concatenation* combines two strings using the _____ operator. (2)

   A. `%`
   B. `+`

True or False: Evaluating boolean expressions

9. `not (3 + 5.1 > 16 / 2)` (2)

   A. True

   B. False

10. `"42" == 42` (2)

    A. True

    B. False

11. `"A" < "B"` (2)

    A. True

    B. False

12. `not (True and False)` (2)

    A. True

    B. False

13. `15 % 4 == 15 / 4` (2)

    A. True

    B. False

Multiple Choice: Reading code

14. What is the *value* in `x` after the following line of code is executed? (4)

```
x = ("a" + "b")[1]
```

    A. `"a"`

    B. `"+"`

    C. `"b"`

    D. This code causes an error; there is no character with index 1.

15. What is the *value* in `x` after the following line of code is executed? (4)

```
x = (3.0 + 1.0) ** (1/2)
```

A. `4.0`

B. `2.0`

C. `1.0`

D. `0.0`

16. What is the *error* produced when the following function is run, given that there is no other code in the file? (4)

```
def question():
    return x * 5.0
```

A. This code does not cause an error.

B. `IndexError: string index out of range`

C. `TypeError: can't multiply sequence by non-int of type 'float'`

D. `NameError: global name 'x' is not defined`

17. Which of the following is a correct call to the function `stars`, defined here? (4)

```
def stars(num):
    """ Prints out num stars separated by spaces """
    count = 0
    while count < num:
        print "*",
        count += 1
```

A. `stars`

B. `stars(100)`

C. `stars[5]`

D. `stars()`

18. What is the *data type* of `x` after the following line of code is executed? (4)

```
x = len( "$15.75"[0] )
```

A. `str` (string)

B. `float`

C. `bool` (boolean)

D. `int` (integer)

19. What is the *data type* of `x` after the following line of code is executed, given that the user enters `-5.7` at the prompt? (4)

```
x = input("Enter a number:")
```

A. `NoneType`

B. `float`

C. `bool` (boolean)

D. `int` (integer)

20. What is the value in `x` after the following code executes? (4)

```
def fcn(a):
   print a / 5

x = fcn(4)
```

A. `0`

B. `0.8`

C. `a`

D. `None`, the function does not return a value.

21. What is the *value* in `x` after the following line of code is executed? (4)

```
x = "hello world" - "l"
```

A. This code causes an error; the `-` operator cannot be used with strings.

B. `"helo world"`

C. `"heo word"`

D. `"hello worldl"`

22. What is the *output* produced after the following code is executed? **Be careful!** (4)

```
num = 75

if num > 10:
   print "Room Alpha"
if num > 50:
   print "Room Bravo"
if num > 100:
   print "Room Charley"
else:
   print "Room Delta"
```

A. `Room Alpha`
   `Room Bravo`

B. `Room Alpha`

C. `Room Alpha`
   `Room Bravo`
   `Room Delta`

D. `Room Bravo`

Fill-in-the-blank: Writing code

For each of the following questions, fill in the value, operator, or statement needed to produce the indicated output (check the comments if you need a hint). Pay attention to data types!

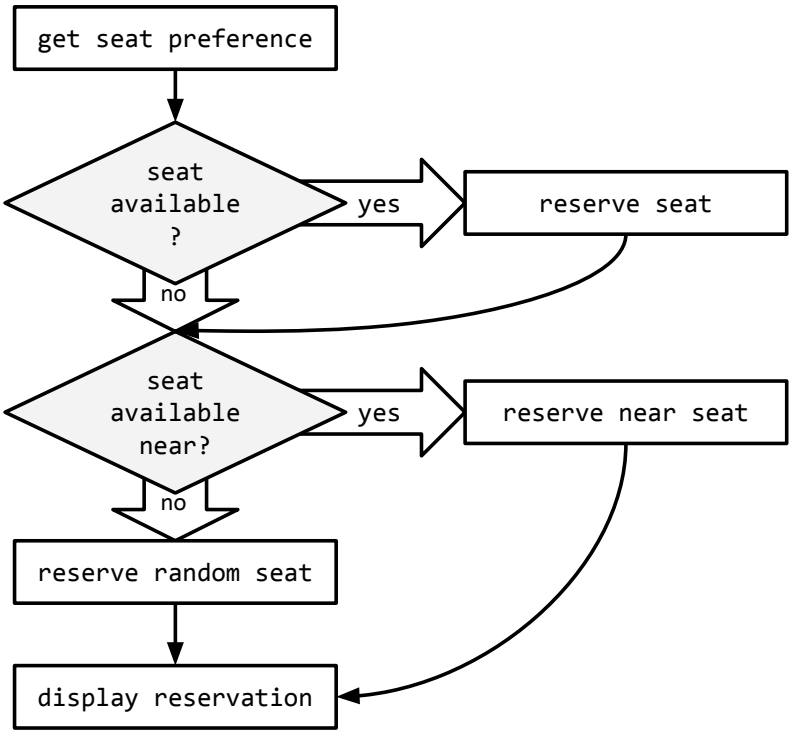
23. Fill in the blank in the condition so that it evaluates to `False` for the `int` values of `x` indicated on the number line below. Empty circles are not included in the range. (6)

`x != -2 and (` _____ `)`

24. Fill in the piece of the following counter-controlled loop so that it will print the word `hi` **three (3)** times. (6)

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

25. Your manager at a theater has provided you with a flowchart representing the logic in your new seat reservation app. Each blank in this question is worth 4 points. (12)



The manager has also written a bit of the code using some functions, but isn't quite sure how to set up the logic of when to call them so that it matches the flowchart. Help your manager finish this code fragment using your knowledge of Python conditions.

```
seat = input("Where would you like to sit?")
__________ available(seat):
     reserve(seat)                        # reserves an available seat
__________ available_near(seat) != None:
     reserve(available_near(seat))        # reserves a seat nearby
__________ :                            # manager note:  do i need this???
     reserve(random_seat())               # reserves a random seat
print get_reservation()
```

This page intentionally left blank.

Please leave it attached to your exam.

