

# CS 320 Exam 3 (15%) - Spring 2022

Instructor: Tyler Caraza-Harter

First/Given Name: \_\_\_\_\_. Last/Surname: \_\_\_\_\_

Net ID: \_\_\_\_\_@wisc.edu

Fill in these fields (left to right) on the scantron form (use 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 A of SPECIAL CODES, write your lecture number, fill in bubbles. 1=8:50am, 2=11am
4. Under B of SPECIAL CODES, tell us about the nearest person (if any) to your left. 0=no person to the left in your row, 1=somebody you do not know is there, 2=somebody you do know is there.
5. Under C of SPECIAL CODES, do the same as B, but for the person to your right
6. **Under D of SPECIAL CODES, write 8 and fill in bubble 8.** This is very important!

Make sure you fill all the special codes above accurately in order to get graded.

You have 2 hours to take the exam. Use a #2 pencil to mark all answers. When you're done, please hand in these sheets in addition to your filled-in scantron. You may not sit adjacent to your friends or other people you know in the class (having only one empty seat is considered "adjacent"). You may only reference your notesheet. You may not use books, your neighbors, calculators, or other electronic devices on this exam. Turn off and put away portable electronics now. If multiple answers to a question are correct, choose the best answer.

(Blank Page for You to Do Scratch Work)

**Q1. Assume the following code runs successfully on some unknown string. What letter(s) could possibly be printed?**

```
matches = re.findall(r"(A)B([CD])", some_string)
print(matches[-1][0])
```

(A) A only    (B) B only    (C) C only    (D) C or D    (E) A, C, or D

**Q2. Every non-leaf node in a binary search tree has at most two children.**

(A) True    (B) False

**Q3. What does `nums` contain after the following runs?**

```
nums = []
def h(z):
    if z > 2:
        h(z-1)
    nums.append(z)
h(5)
print(nums)
```

(A) [2, 3, 4]    (B) [2, 3, 4, 5]    (C) []    (D) [4, 3, 2]    (E) [5, 4, 3, 2]

**Q4. True or False? The B class is a child of the A class; both have an `__init__` method. Both `__init__` methods are guaranteed to run when a new instance of B is created, regardless of the code in B's `__init__` method.**

(A) True    (B) False

**Q5. True or False. A generator object returned by a generator function supports indexing, slicing, and looping, just like a list.**

(A) True    (B) False

**Q6. What is an advantage that a typical GPU has over a typical CPU?**

(A) more cores    (B) more flexible cores    (C) cores with a faster clockrate    (D) all of the above

**Q7. If we can successfully call `obj.fit(df)`, what type could `obj` be?**

(A) LinearRegression only  
(B) LinearRegression or LogisticRegression  
(C) KMeans only  
(D) KMeans or AgglomerativeClustering

**Q8. In complexity analysis, which statement about steps is true?**

- (A) all steps must take the same amount of time to execute
- (B) a step's execution time never depends in any way on the input
- (C) a piece of code containing a loop always consists of multiple steps
- (D) a step has a bounded execution time that doesn't keep growing with input size

**Q9. What is the recall for oranges, given the following confusion matrix?**

	apples	oranges	bananas
apples	4	2	3
oranges	5	3	2
bananas	4	0	5

- (A) 0.3   (B) 0.6   (C) 3   (D) 10

**Q10. Does the regular expression `r"a.*?b"` match anything in the string `".bdc1Ba"`?**

- (A) yes   (B) no

**Q11. How many oranges are classified as apples, according to the following confusion matrix?**

	apples	oranges	bananas
apples	83	99	49
oranges	18	82	58
bananas	51	45	88

- (A) 18   (B) 81   (C) 99   (D) 117

**Q12. What is a valid simplification of the following, assuming the code runs without error?**

```
X @ np.linalg.solve(X, y)
```

- (A) `x`   (B) `y`   (C) `c`   (D) `x @ y`

**Q13. What is `x`?**

```
class Fruit:
    def __init__(self, vals):
        self.vals = vals

    def __len__(self):
        return 2

    def __getitem__(self, lookup):
        return 1
```

```
obj = Fruit([5, 3, 4])
x = len(obj.vals) # careful!
```

(A) 1   (B) 2   (C) 3   (D) 4   (E) 5

**Q14. `df` has 13 columns and 156 rows. After running the following, how many columns will `p.components_` have?**

```
p = PCA(4)
p.fit(df)
```

(A) 4   (B) 13   (C) 52   (D) 156

**Q15. If a Flask app has the following handlers, what does it print when a user visits the home page in a browser?**

```
@app.route("/")
def home():
    print("X")
    return '<html><body></body></html>'

@app.route("/example.svg")
def f():
    print("Y")
    return "TODO"

@app.route("/out.svg")
def g():
    print("Z")
    return "TODO"
```

(A) X only   (B) X and Y   (C) X and Z   (D) X, Y, and Z

**Q16. You want to create a model to predict a number. What kind of machine learning task is this?**

(A) regression   (B) classification   (C) clustering   (D) decomposition

**Q17. You want to extract a subset of a DataFrame's columns to produce a new DataFrame, to which you'll fit KMeans. What should go between the brackets?**

- (A) list of strings
- (B) a make\_column\_transformer
- (C) a string, equal to the name of the label column
- (D) a string, equal to the name of a feature column

**Q18. If a BST is constructed using the algorithm we learned in class, and the insert order is [9, 13, 4, 1], where will 1 be?**

- (A) root.left.left
- (B) root.left.right
- (C) root.right.left
- (D) root.right.right

**Q19. If `A=np.array([[6, 4], [2, 5]])` and `b=np.array([[3, 1]])`, what is `A*b`?**

- (A) `[[18,12],[2,5]]`
- (B) `[[18,4],[6,5]]`
- (C) `[[22],[11]]`
- (D) `[[24,16],[8,20]]`

**Q20. Is the center of the text created by the following code within the region bounded by the ax subplot?**

```
ax.text(0.1, 0.9, "hi", transform=ax.transAxes, size=20, color="red",
        verticalalignment="center", horizontalalignment="center")
```

- (A) definitely not
- (B) definitely
- (C) it depends on the xlim and ylim of the ax region

**Q21. A square with alpha 1 is hidden beneath a triangle. If you want to see the square, what could you do?**

- (A) increase alpha of the square
- (B) decrease zorder of the triangle
- (C) increase alpha of the triangle
- (D) increase zorder of the triangle
- (E) decrease zorder of the square

**Q22. If you want to produce a dendrogram, what should you use?**

- (A) LinearRegression
- (B) LogisticRegression
- (C) KMeans
- (D) AgglomerativeClustering
- (E) PCA

**Q23. What is the complexity of the following code, if N is the length of the list L? Choose the best answer.**

```
for num in L:
    threshold = min(L)
    if num > 4 * threshold:
        print("oops!")
```

- (A)  $O(1)$
- (B)  $O(N)$
- (C)  $O(N^2)$
- (D)  $O(N^2 + 1)$
- (E)  $O(N^3)$

**Q24. What is printed?**

```
h = []
for item in [1.5, 0, -5, 2, -1]:
    heapq.heappush(h, item)
print(heapq.heappop(h))
```

- (A) -1   (B) -5   (C) 0   (D) 1.5

**Q25. In the following code analyzing a contingency table from an A/B test, pvalue is 0.022. The threshold for significance is 0.02. Do we have statistically significant evidence that B has a different click-through-rate than A?**

```
from scipy import stats

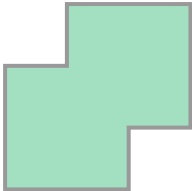
df = pd.DataFrame({
    "click": {"A": ????, "B": ????,},
    "no-click": {"A": ????, "B": ????,},
})

_, pvalue = stats.fisher_exact(df)
```

- (A) yes   (B) no

**Q26. What could be added to the following code to produce the below shape?**

```
from shapely.geometry import box, Point
x = box(0, 0, 2, 2)
y = box(1, 1, 3, 3)
```



- (A) `x.union(y)`   (B) `x.intersection(y)`   (C) `x.difference(y)`   (D) `y.difference(x)`

**Q27. The shape of A is (7, 3), the shape of B is (3, 1), and the shape of C is (1, 9). What is the shape of A@B@C?**

- (A) (1, 9)   (B) (3, 1)   (C) (7, 3)   (D) (7, 9)

**Q28. What scikit-learn transformer could have potentially produced the following output features?**

```
array([[ 1.,  1.,  0.,  1.,  0.,  0.],
       [ 1., -1.,  1.,  1., -1.,  1.]])
```

- (A) PolynomialFeatures   (B) OneHotEncoder   (C) StandardScaler

**Q29. What are the bounds on possible values for `s`?**

```
lr = LogisticRegression()  
lr.fit(train[xcols], train[ycol])  
s = lr.score(test[xcols], test[ycol])
```

(A) -infinity to infinity    (B) -infinity to 1    (C) 0 to 1    (D) 0 to infinity    (E) -1 to 1

**Q30. What call makes predictions using a computation similar to `x @ c`?**

(A) `LinearRegression.predict`    (B) `LogisticRegression.predict`  
(C) `LinearRegression.predict_proba`    (D) `LogisticRegression.predict_proba`