Q1 SQL and Relational Algebra
8 Points

Q1.1
1 Point

Consider the following SQL query. Which of the following relational algebra expressions will produce an equivalent result?

```
SELECT a.name, b.instructor_for
FROM student AS a, teachers AS b
WHERE a.class = b.instructor_for AND b.area = “datascience”
```

- ○ $\pi_{name,instructor\_for}(\sigma_{class=instructor\_for}(student \times teachers))$
- ○ $\pi_{name,instructor\_for}(\sigma_{class=instructor\_for}(student) \times \sigma_{b.area=\text{“}datascience\text{”}}(teachers))$
- [✓] $\pi_{name,instructor\_for}(\sigma_{class=instructor\_for}(student \times \sigma_{b.area=\text{“}datascience\text{”}}(teachers)))$
- ○ $\sigma_{class=instructor\_for}(student \times \sigma_{b.area=\text{“}datascience\text{”}}(teachers))$

Q1.2
1 Point

For each of the following relational algebra operators, which is the relation between the total number of rows (if there are multiple inputs/outputs, add them together) in the input and the output?

Note: answer in the general case (including empty relations). Select the most precise correct answer (equality is more precise)

π

- ○ $|input| \geq |output|$
- ○ $|input| \leq |output|$
- [✓ $|input| = |output|$]
- ○ None of the above

Q1.3
1 Point

σ

- [✓ $|input| \geq |output|$]
- ○ $|input| \leq |output|$
- ○ $|input| = |output|$
- ○ None of the above

Q1.4
1 Point

$\times$

- ○ $|input| \geq |output|$
- ○ $|input| \leq |output|$
- ○ $|input| = |output|$
- [✓ None of the above]

Q1.5
1 Point

Lakshya is building a new web application for tracking election analyses, and wants to perform a couple queries. For each query, select the appropriate SQL logic to put in the blank.

The database under the hood has a relatively simple schema:

```
Polls(INT measure, TEXT state, TEXT county, INT yes_votes, INT estimated_voters, BOOL estimated_win)

Measures(INT id, TEXT proposed_by, INT money_required)
```

Note: there may be multiple polls in the same location corresponding to different data sources

Lakshya wants to start by calculating the total number of counties from California that have data available.

```
SELECT ________
FROM Polls
WHERE ________
```

Blank 1:

○ COUNT(*)

[✓ COUNT(DISTINCT county)]

○ SUM(*)

Blank 2:

[✓ state = “California”]

○ measure = id AND state = “California”

○ state.id = ID_CALIFORNIA

Q1.6
1 Point

Next, Lakshya wants to compute the average estimated voters across the data sources for each county.

```
SELECT state, county, AVG(estimated_voters)
FROM Polls
____________
```

- ○ WHERE county IS UNIQUE
- ○ WHERE state, county IS UNIQUE
- ○ GROUP BY county
- [✓ GROUP BY state, county]

Q1.7
0.5 Points

Lakshya is a bit suspect of the number of measures proposed by a “David Chu” that are asking for money to purchase swag for the Sky Lab. Help him write a query that counts the amount of money requested for each measure by “David Chu” that has at least 50 polls with an estimated win.

Note: there is exactly one row for each measure in the Measures table

```
SELECT measure.id, measure.money_required
FROM ____________________
___________________
___________________
___________________
```

Blank 1:

- ○ Measures AS measure
- [✓ Measures AS measure INNER JOIN Polls AS poll ON measure.id = poll.measure]
- ○ Measures AS measure, Polls AS poll

Q1.8
0.5 Points

Blank 2:

- [✓ WHERE measure.proposed_by = “David Chu”]
- WHERE measure.proposed_by = “David Chu” AND COUNT(poll) >= 50
- WHERE measure.proposed_by = “David Chu” AND COUNT(poll.estimated_win = TRUE) >= 50

Q1.9
0.5 Points

Blank 3:

- [✓ GROUP BY measure.id, measure.money_required]
- GROUP BY measure.id
- GROUP BY poll.state, poll.county
- no GROUP BY required

Q1.10
0.5 Points

Blank 4:

- HAVING COUNT(poll) >= 50
- [✓ HAVING COUNT(poll.estimated_win = TRUE) >= 50]
- no HAVING required

Q2 Indexes
3 Points

Q2.1
1 Point

B+ trees require periodic manual rebalancing in order to ensure efficient retrieval of data.

○ True

[✓ False]

Q2.2
1 Point

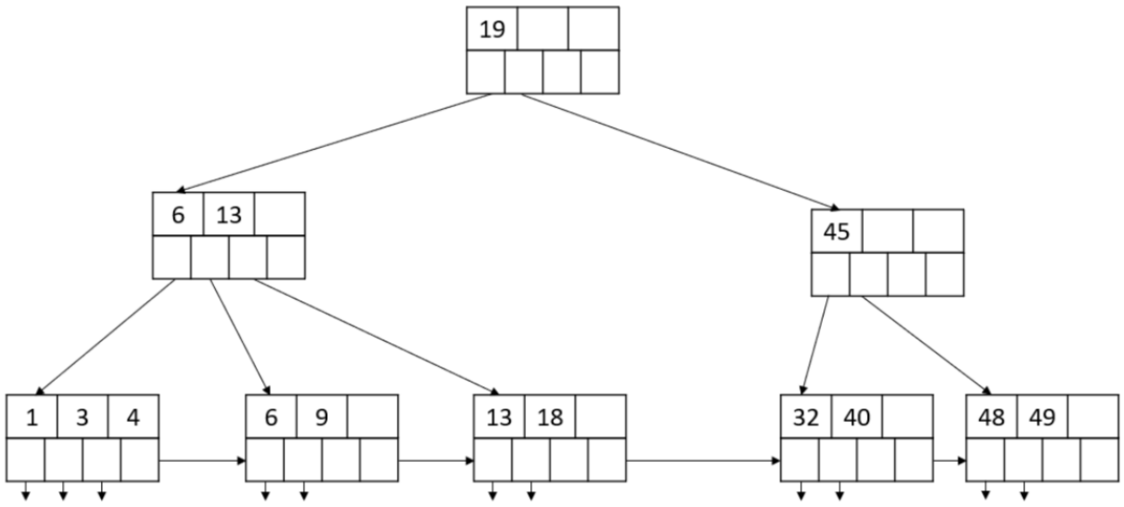
Using an index for a query will always be at least as efficient as using a sequential scan.

○ True

[✓ False]

Q2.3
1 Point

Consider the following B+ tree from discussion.



Assuming we want to do a range search between [19, 48], how many tree nodes would we touch in the process?

4

Q3 Query Optimization
6 Points

Consider the table students, with 1,000 pages and 1 million tuples, and an index with height = 4 on students.name.

- 90% of the students in the table have a GPA above 2.0 and 50% of students have a GPA above 3.0.
- No more than 10 students share the same name.
- You are given no other information regarding this table.
- (Reminder: height is the number of pointers you must traverse to get from the root to the leaf)

```
CREATE TABLE students (
  id INT PRIMARY KEY,
  name VARCHAR[11] NOT NULL,
  major VARCHAR[9] NOT NULL,
  GPA FLOAT NOT NULL,
);
```

Q3.1
1 Point

If we were to execute the query `SELECT * from students where GPA > 3.0`, how many pages of students would we need to read, in the worst case?

1000

Q3.2
1 Point

If we were to execute the query `SELECT * from students where name = ‘John’ or GPA > 3.0`, how many pages of students would you need to read, in the worst case?

1000

Q3.3
0.5 Points

Now assume the university decides to track the student athletes specially. There is now a second table `student_athletes` with 100 pages and 10,000 tuples, to go along with the `students` table.

```
CREATE TABLE student_athletes (
  id INT PRIMARY KEY,
  name VARCHAR[11],
  sport VARCHAR[9],
  student_id INT FOREIGN KEY references students.id
);
```

Now, assume we have three indexes in total.

```
students: name
student_athletes: name, sport (2 separate indexes)
```

Consider the following query:

```
SELECT student_athletes.sport, AVERAGE(gpa),
FROM students INNER JOIN student_athletes
ON students.id = student_athletes.student_id WHERE student_athletes.sport = 'soccer'
GROUP BY student_athletes.sport;
```

To execute this query, we need to read the `students` and `student_athletes` tables. Put the following operations in the order in which they will be performed, starting from 1. If they will not be performed in this query, put `N/A`.

If two operations happen concurrently, give them the same number n and give the next operation the number n+1 (e.g do not "skip" a number).

The following 6 blanks (Q3.3 to Q3.8) are all part of this question.

Filtering on sport = 'soccer'

1

Q3.4
0.5 Points

Index scan on student_athletes.sport

1

Q3.5
0.5 Points

Index scan on students.name

N/A

Q3.6
0.5 Points

Averaging the GPA

4

Q3.7
0.5 Points

Join student_athletes and students

2

Q3.8
0.5 Points

Group by student_athletes.sport

3

Q3.9
1 Point

What are some optimization strategies the optimizer may decide to use?

- [✓] Push down the `student_athletes.sport = 'soccer'` filter to be before the join
- [ ] Perform an index join between `student_athletes` and `students`.
- [ ] Project away the columns not in the SELECT clause when scanning `student_athletes` and `students`
- [ ] None of the above

Q4 Data Imputation

9 Points

Let us assume our set of numbers is: [1, 20, 30, 40, 100].

Round your answer to the nearest hundredth decimal if your answer is not exact.

Q4.1

1 Point

Calculate the mean.

38.2

Q4.2

1 Point

Calculate the median.

30

Q4.3

1 Point

Calculate the population standard deviation.

33.48

Q4.4

1 Point

Calculate the median absolute deviation (MAD).

10

Q4.5
1 Point

If we determine outliers as points that lie 2 standard deviations away from the mean, which points are outliers?

- ☐ 1
- ☐ 20
- ☐ 30
- ☐ 40
- ☐ 100
- [✓ None of the above]

Q4.6
2 Points

If we want to winsorize the data using a 50% winsorization, which values would get modified? Assume that for the purposes of this question, the 25th percentile value is 20 and the 75th percentile value is 40.

- [✓ 1]
- ☐ 20
- ☐ 30
- ☐ 40
- [✓ 100]
- ☐ None of the above

Q4.7
2 Points

Input your new array, **in sorted order (ascending)**, after the 50% winsorization applied in the previous part. Your answer should follow the exact format `[A, B, C, D, E]` (e.g. `[10, 20, 30, 40, 50]`). (Credit will not be awarded if this format is not followed.)

[20, 20, 30, 40, 40]

Q5 MDL and String Functions

4 Points

Assume you have a data column with values {'Ball', 0, 1, 300, 5}. The default encoding type is `string` (8 bits per character).

Q5.1

1 Point

What is the MDL for encoding this as `int16` (16 bits per integer)?

96

Q5.2

1 Point

What is the MDL for encoding this as a `string` (8 bits per character)?

80

Q5.3

1 Point

What would MDL favor, in this case?

○ `int16`

[✓ `string`]

Q5.4 Levenshtein Distances

1 Point

Calculate the Levenshtein distance between "golf" and "ball"

3

Q6 Transactions
9.5 Points

Q6.1
1 Point

A schedule that is __ is also ____.

- ○ Serializable, Conflict Serializable
- [✓ Conflict Serializable, Serializable]
- ○ Conflict Serializable, Durable
- ○ Atomic, View Serializable

Q6.2
1 Point

In Two Phase Locking, the two phases are called...

- ○ Write and Commit
- ○ Abort and Commit
- [✓ Acquire and Release]
- ○ Read and Write

Q6.3
1 Point

A non-serializable schedule violates which ACID property?

- ○ Atomicity
- ○ Durability
- ○ Consistency
- [✓ Isolation]
- ○ None of the above

Q6.4
1 Point

A transaction T1 begins running, writes to page P1, and then commits. Following this, the database crashes. When it is rebooted, T1's writes are gone. Which ACID property does this violate?

- ○ Atomicity
- [✓ Durability]
- ○ Consistency
- ○ Isolation
- ○ None of the above

Q6.5
1 Point

A transaction T2 runs concurrently with transaction T1. T2 writes to page P2. T1 writes to page P1. Which ACID property does this violate?

- ○ Atomicity
- ○ Durability
- ○ Consistency
- ○ Isolation
- [✓ None of the above]

Q6.6
1 Point

A transaction T1 writes to page P1. The database crashes before T1 completes, and when it reboots, T1’s partial writes are still present on P1. Which ACID property does this violate?

- [✓ Atomicity]
- Durability
- Consistency
- Isolation
- None of the above

Q6.7
1 Point

Assume we are using Strict Two Phase Locking, and transaction T1 reads page P1. Transaction T1 then writes to page P1. Are the two actions in conflict?

- Yes
- [✓ No]

Q6.8
1 Point

Transaction T1 writes to page P1. Transaction T2 then reads P1 and writes its values to page P2. T1 then writes to P2 before issuing a Rollback to abort. If using strict Two Phase Locking, this would result in...

- An abort of T1 and T2 both (a cascading abort)
- A Blind Write
- A Dirty Read
- [✓ This is impossible under strict 2PL.]

Q6.9
1.5 Points

Time	1	2	3	4	5	6	`
T1	W(A)	-	-	R(C)	-	-	
T2	-	R(A)	-	-	-	W(C)	
T3	-	-	R(A)	-	W(B)	-	

Is the above schedule conflict serializable? If not, please put in the conflicting transactions involved that make it non-conflict serializable, ordered lexicographically and separated by commas (e.g. `T1, T2`). If it is, please write "Conflict Serializable".

Conflict Serializable

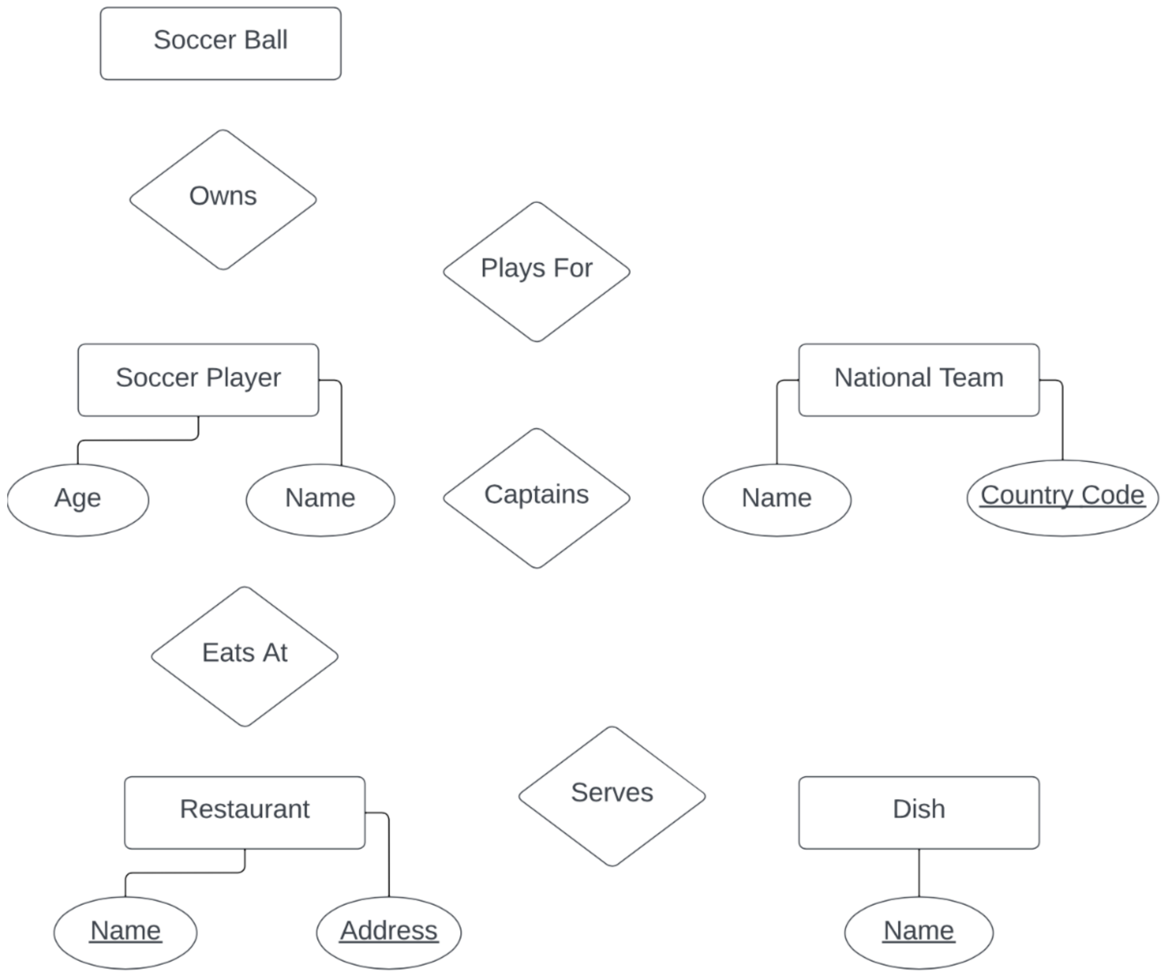
Q7 ER Diagrams

5 Points

It’s World Cup Time! We want to draw an Entity-Relationship diagram describing the following real-world scenario.

- A player owns at least one soccer ball, and each soccer ball only has one owner
- A soccer player may play for at most one national team, and each national team must have at least one player.
- A player may captain one team but is under no obligation to captain at all. Each national team must have exactly one captain.
- A player may choose to eat at restaurants while they play in the World Cup, but are under no obligation to do so. Every restaurant can have as many players eating at it, or none at all.
- Each restaurant, identified by its address and name, has to serve multiple dishes. Each dish is uniquely identified by its name and the restaurant serving it.

You are given the following diagram to guide you. Your role is to fill it out with the appropriate lines.



Q7.1 What line should be drawn between the Soccer Player entity and Captains relationship set?
0.5 Points

[✓ Thin Arrow]

Thick Arrow

Thin Line

Thick Line

Q7.2 What line should be drawn between the Captain relationship set and National Team entity?
0.5 Points

- Thin Arrow
- [✓ Thick Arrow]
- Thin Line
- Thick Line

Q7.3 What line should be drawn between the Soccer Player entity and Plays For relationship set?
0.5 Points

- [✓ Thin Arrow]
- Thick Arrow
- Thin Line
- Thick Line

Q7.4 What line should be drawn between the National team entity and Plays For relationship set?
0.5 Points

- Thin Arrow
- Thick Arrow
- Thin Line
- [✓ Thick Line]

Q7.5 What line should be drawn between the Soccer Ball entity and Owns relationship set?
0.5 Points

- Thin Arrow
- [✓ Thick Arrow]
- Thin Line
- Thick Line

Q7.6 What line should be drawn between the Soccer Player entity and Owns relationship set?
0.5 Points

- Thin Arrow
- Thick Arrow
- Thin Line
- [✓ Thick Line]

Q7.7 What line should be drawn between the Soccer Player entity and Eats at relationship set?
0.5 Points

- Thin Arrow
- Thick Arrow
- [✓ Thin Line]
- Thick Line

Q7.8 What line should be drawn between the Restaurant entity and Eats at relationship set?
0.5 Points

- Thin Arrow
- Thick Arrow
- [✓ Thin Line]
- Thick Line

Q7.9 What line should be drawn between the Dish entity and Serves relationship set?
0.5 Points

- Thin Arrow
- [✓ Thick Arrow]
- Thin Line
- Thick Line

Q7.10 What line should be drawn between the Restaurant entity and Serves relationship set ?
0.5 Points

- Thin Arrow
- Thick Arrow
- Thin Line
- [✓ Thick Line]

Q8 NoSQL
9 Points

Q8.1 Select all that are true about NoSQL
2 Points

- [✓] While SQL databases follow ACID principles and guarantee (strong) consistency, NoSQL databases only have to guarantee eventual consistency.
- [ ] SQL databases are more horizontally scalable than NoSQL databases because they can be more easily distributed across different data stores.
- [ ] Scaling through partitioning helps improve fault tolerance.
- [✓] Document stores such as MongoDB and key value stores such as DynamoDB are both examples of NoSQL databases.

Q8.2 Select all that are true about JSON
2 Points

- [ ] Modern relational databases such as Postgres cannot support semi-structured data such as JSON
- [✓] JSON can be easily manipulated by Javascript in the web browser
- [✓] One advantage of JSON is that we can add a new attribute for new tuples without modifying other tuples
- [ ] JSON Document Store is a data model for structured data

Q8.3 Select all that are true about Mongo
2 Points

- ☐ The Mongo `$lookup` operator is most similar to an inner join in Postgres.
- ☐ Similar to Postgres, Mongo also has access to different kinds of joins including hash join and sort merge join.
- ☐ Mongo retrieval `find` query can be used to perform group by operations.
- [✓ Mongo aggregation queries capture retrieval queries as a special case (e.g. anything that can be done with a retrieval `find` query can also be done with an aggregation query).]

Q8.4 Which of the following best describes MapReduce?
1 Point

- ○ MapReduce is a database query language used to perform complex operations on large datasets.
- [✓ MapReduce is a parallel computing framework used to process large amounts of data across a distributed cluster of machines.]
- ○ MapReduce is a database management system used to store and manage large amounts of structured data.
- ○ MapReduce is a programming model used to write algorithms that can be executed on a distributed cluster of machines.

Q8.5 Select all use cases that are more suited for NoSQL databases as opposed to SQL databases
2 Points

- [ ] Applications that require complex queries and transactions, such as data warehouses and online analytical processing (OLAP) systems
- [✓] Applications that require fast access to data, such as real-time gaming applications
- [ ] Applications where data consistency and integrity are critical, such as financial and government systems
- [✓] Applications that store large amounts of semi-structured data, such as web logs and social media data

Q9 MongoDB

8.5 Points

Note: keep in mind for this question that we are not testing on syntax here, so don't worry about any syntax errors (e.g. missing parentheses) if they exist.

Shadaj is working on assigning final grades to students in Data 101, and would like your help! All student data are stored in the students collection in MongoDB, with each document containing the studentId, name, and score fields (that always exist).

Q9.1

2 Points

Shadaj would like to assign a grade of "A" to students with scores in the range of 93 (inclusive) and 99 (exclusive). Which of the following Mongo queries accomplishes this?

- [✓ db.students.updateMany({ score: { $gte: 93, $lt: 99 } }, { $set: { grade: "A" } })]
- db.students.updateMany({ $set: { grade: "A" } }, { score: { $gte: 93, $lt: 99 } })
- [✓ db.students.updateMany({ $and: [{ score: { $gte: 93 } }, { score: { $lt: 99 } }], { $set: { grade: "A" } })]
- db.students.updateMany({ $set: { grade: "A" } }, { $and: [{ score: { $gte: 93 } }, { score: { $lt: 99 } }])

Q9.2

1 Point

How would you write the above query in SQL?

- [✓ UPDATE students SET grade = 'A' WHERE score >= 93 AND score < 99;]
- UPDATE students WHERE score >= 93 AND score < 99 SET grade = 'A';
- SET grade = 'A' UPDATE students WHERE score >= 93 AND score < 99;
- SELECT grade = 'A' FROM students WHERE score >= 93 AND score < 99;

Q9.3
1 Point

MapReduce is commonly used with NoSQL databases such as MongoDB. Now, suppose Shadaj wants to count the number of students that have achieved the grade of “A”. Select the option that best define the map and the reduce functions using MapReduce paradigm to achieve the objective.

- ○ [Map function] `lambda score: return score if grade == “A” else 0`; [Reduce function] `lambda scores: return sum(scores)`
- [✓ [Map function] `lambda score: return 1 if grade == “A” else 0`; [Reduce function] `lambda scores: return sum(scores)`]
- ○ [Map function] `lambda score: return 1`; [Reduce function] `lambda scores: return sum([score for score in scores if grade == ‘A’])`
- ○ [Map function] `lambda score: return score`; [Reduce function] `lambda scores: return sum([score for score in scores if grade == ‘A’])`

Q9.4
1.5 Points

Consider that we have a collection `scores` that contain the following documents.

```
{score: 90, students: [1, 2, 3, 4] }
{score: 80, students: [5, 6] }
{score: 70, students: [7, 8, 9] }
```

How many output documents do we have after running the following query? The answer should be an integer.

```
db.scores.aggregate( [
{ $match: { score: { $gte: 80 } } },
{ $unwind: "$students"}
] );
```

6

Q9.5
1.5 Points

We want to store the output of the previous query on a distributed cluster of machines. Assume that we perform a hash-based partitioning by the score field, and we replicate each document so that it is stored on three machines. Furthermore, assume that the hash functions are perfect with no key collisions (e.g. each key is mapped to a distinct value).

How many hash functions are needed?

3

Q9.6
1.5 Points

Continuing from the previous question...

How many machines are needed?

6

Q10 Academic Integrity
0 Points

I followed all exam procedures for this exam. I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct and may further result in partial or complete loss of credit.

[✓ Yes]