CIS 1100—Practice Exam I, Spring 2026

Name: _____ PennID (e.g. 12345): _____

My signature below certifies that I have complied with the University of Pennsylvania's Code of Academic Integrity in completing this examination.

_____ _____

Signature Date

- Do not open this exam until told by the proctor.
- You will have exactly 60 minutes to take this exam. There are five questions plus a bonus.
- Make sure your phone is turned OFF (not on vibrate!) before the exam starts.
- Food and gum are not permitted—don't be noisy or messy.
- You may not use your phone or open your bag for any reason, including to retrieve or put away pens or pencils, until you have left the exam room.
- This exam is closed-book, closed-notes, and closed computational devices.
- If you get stuck on a problem, it may be to your benefit to move on to another question and come back later.
- All code must be written in proper Python format.
- Do not separate the exam pages. Do not take any exam pages with you. The entire exam packet must be turned in as is.
- There are blank pages at the end of the exam if you need extra space for any graded answers.
- Use a pencil or a dark pen to complete the exam.
- If you have any questions, raise your hand and a proctor will come to you.
- When you turn in your exam, you may be required to show your PennCard. If you forgot to bring your ID, talk to an exam proctor immediately.
- We wish you the best of luck!

Q1. Types

In the column marked **"Type,"** choose the type (`int`, `float`, `bool`, `str`, `list`, `tuple`, `range`) for the expression, or write **"error"** if there is an error in the expression. You do not need to write the value of the expression. In cases where multiple answers are possible, any of them will be accepted.

Code	**Type**
`28 % 7 == 1`	
`3`	
`"3"`	
`"3" in [3]`	
`[4] + [5]`	
`[4] + 5`	
`ord("c")`	
`"c" + 67`	
`4 < 3 <= 2`	
`[1, 2, 3][3]`	

Q2. Values

Write the value that gets printed, or write **“error”** if there is an error during the execution of these lines of the program. **Hint:** remember that **None** is a value.

Question 2.1

```
h = "bird"
h = "plane"
print(f"It's a {h}! It's a {h}!")
```

Answer:

Question 2.2

```
k = [7, "tree", True]
for thing in range(k):
    print(thing)
```

Answer:

Question 2.3

```
t = 18 // 9
r = 18 % 9
print(t - r)
```

Answer:

Question 2.4

```
print(10 + len("10"))
```

Answer:

Question 2.5

```
x = [4, 5, 6, 7]
x[4] = 8
print(x)
```

Answer:

Question 2.6

```
j = []
j.append([4, 5, 6])
j.append(5)
print(j)
```

Answer:

Question 2.7

```
e = [1, 2, 3, 4, 5]
e[0] = 5
print(len(set(e)))
```

Answer:

Question 2.8

```
def mystery(start, text):
    msg = ""
    while start > 0:
        msg = msg + str(start) + "..."
        start -= 1
    return msg + f"{text}"

print(mystery(3, "Go!"))
```

Answer:

Q3. Debugging: Serial Number Decoder

This page contains only instructions; the place for you to write will be on the next page.

In order to share the origin of an electric guitar, the manufacturer typically prints a serial number somewhere on the back of it. That serial number encodes several pieces of information:

- **country of origin**, encoded in the first character as `'0'` for Mexico, `'1'` for Japan, and `'2'` for USA
- **brand:** if the second character is `'S'`, the guitar is a *Squier* brand; if it is an `'F'`, it's a *Fender*.
- **month** and **year**: the next six characters are always related to the date of manufacture:
  - Fender guitars write the date as YYYYMM, e.g. `201608` for August 2016
  - Squier guitars write the date as MMYYYY, e.g. `082016` for August 2016
- **model number** between `000` and `999` indicates what kind of guitar it is:
  - even numbers between `0-200` (inclusive) are for Jazzmasters, odd numbers are for Jaguars
  - `201-550` (inclusive) for Stratocasters
  - `551-999` (inclusive) for Telecasters

Serial Number	Expected Output
"0F201409101"	("Mexico", "Fender", "2014", "09", "Jaguar")
"2S102009550"	("USA", "Squier", "2009", "10", "Stratocaster")
"1F202412004"	("Japan", "Fender", "2024", "12", "Jazzmaster")

Fix the bugs in the program on the following page so that it correctly **returns** a tuple of information, i.e. `(country, brand, year, month, kind)`, using the rules described above. You can assume that the serial number is always well-formatted: it will not deviate from these instructions. Identify each of the six lines with bugs and provide corrections for those lines. (There are actually seven bugs, but the first error is solved for you.)

```
fed decode_serial_num(serial: str) -> tuple[str, str, str, str, str]:
    code_mapping = ["Mexico", "Japan", "USA"]
    country = serial[int(code_mapping[0])]

    brand = "Squier"
    if serial[1] == F:
        brand == "Fender"

    if brand == "Fender":
        year = serial[2:5]
        month = serial[6:8]
    else:
        year = serial[2:4]
        month = serial[4:8]

    model_number = int(serial[8:])
    if model_number <= 200 or model_number % 2 == 0:
        kind = "Jazzmaster"
    elif model_number <= 200 and model_number % 2 == 1:
        kind = "Jaguar"
    elif model_number <= 550:
        kind = "Stratocaster"
    else model_number > 550:
        kind = "Telecaster"

    return (country, brand, year, month, kind)
```

Bug #	Line #	Issue/Fix
Bug #1	1	signature is wrong, change fed to def
Bug #2		
Bug #3		
Bug #4		
Bug #5		
Bug #6		
Bug #7		

Q4. Complete the Program: Washer Waitlist

This page contains only instructions; the place for you to write will be on the next page.

The CIS1100 staff has run into a bit of a problem at the dorm! There are only three washing machines but a long line of students are still waiting to do their laundry. The staff decided to write a program to figure out how long someone will have to wait before they can start using a washing machine.

There are three washers, and all of them are available at time 0. Students will use the first washer that becomes free (using the one furthest to the left if more than one is free at the same time). Each student also chooses to either run a **heavy** cycle, which takes 60 minutes, or a **regular** cycle, which takes 45 minutes. Students use the washers in the same order that they appear in the waitlist, from left to right.

Help the TAs complete the **`laundry_timer(waitlist)`** function, where **`waitlist`** is the list of students ahead of the current student in line. The function should return an integer representing the number of minutes until a washer becomes available for the current student.

Example

Calling:

```
waitlist = [
    ("alex", "heavy"),
    ("beth", "regular"),
    ("charlie", "regular"),
    ("diane", "heavy"),
    ("esther", "heavy")
]

laundry_timer(waitlist)
```

Should return the int 60. After assigning everyone in **`waitlist`**, the machines' completion times are [60, 105, 105]. So the next student will be able to start after 60 minutes.

Incomplete Program

```
def laundry_timer(waitlist: list[tuple[str, str]]) -> int:
    machine_run_times = [0, 0, 0]

    for waiting_tuple in ___BLANK_0___:
      name = waiting_tuple[0]
      kind = waiting_tuple[1]
      min_machine_time = machine_run_times[0]
      min_machine_index = 0

      for index, curr_machine_time in ___BLANK_1___:
        if curr_machine_time < min_machine_time:
          min_machine_time = ___BLANK_2___
          min_machine_index = index

      if kind == "heavy":
        machine_run_times[min_machine_index] += ___BLANK_3___
      else:
        machine_run_times[min_machine_index] += 45

    # find final waiting time based on times per machine
    waiting_time = machine_run_times[0]

    for time in machine_run_times:
      if time < ___BLANK_4___:
        waiting_time = time

    return ___BLANK_5___
```

Fill in the table below to answer this question.

Blank	Code
___BLANK_0___	
___BLANK_1___	
___BLANK_2___	
___BLANK_3___	
___BLANK_4___	
___BLANK_5___	

Q5. Coding: Scorigami

You're a new hire at the National Football League's analytics group. Your job is tracking historical final scores. We can represent a final score as a `tuple` of two ints.

Question 5.1 Standard Form

First, we say that a final score is in **standard form** if the first number is greater than or equal to the second number. `(28, 5)` and `(19, 19)` are scores in standard form; `(5, 28)` and `(14, 28)` are not. You will write a function `standard_scores_only()` that takes in a list of final scores and returns a *copy* of that list with only the scores that are in standard form. Review the examples below, then implement the function.

Input	Expected Output
[]	[]
[(5, 28)]	[]
[(28, 5)]	[(28, 5)]
[(5, 28), (28, 5)]	[(28, 5)]

Question 5.1.1 Implement `standard_scores_only`

```
def standard_scores_only(all_scores: list[tuple]) -> list[tuple]:









    return
```

Question 5.2 Unseen Scores

We say that a score is **unseen** if it is not found in the list of previously observed scores. Write a function `find_unseen_scores()` that takes in a list of final scores from a recent batch of games as well as a list of all previously observed scores. The function should return a `list` of all scores present in the final `list` that are not present in the previously observed `list`, in the order that they appear in the final score `list`.

Recent Final Score List	Previously Observed List	Expected Output
[]	[]	[]
[(28, 5), (15, 4)]	[]	[(28, 5), (15, 4)]
[(28, 5), (15, 4)]	[(15, 4)]	[(28, 5)]

Question 5.2.1 Implement `find_unseen_scores`

```
def find_unseen_scores(
        score_batch: list[tuple],
        previous: list[tuple]
    ) -> list[tuple]:









    return
```

Question 5.3 Count Scorigami

We say that a score is **"scorigami"** if *it is written in standard form and has not previously been observed*. Given a list representing a batch of new scores and a list representing all previously achieved scores, return the number of **"scorigami"** scores in that input list.

For full credit, your solution must call both functions from the previous two parts. You can receive credit even if your implementations for those functions are incorrect.

Recent Final Score List	Previously Observed List	Expected Output
[]	[]	0
[(28, 5), (15, 4)]	[]	2
[(28, 5), (15, 4)]	[(15, 4)]	1
[(5, 28), (15, 4)]	[(15, 4)]	0

Question 5.3.1 Implement `count_scorigami`

```
def count_scorigami(batch: list[tuple], previous: list[tuple]) ->
    int:









    return
```

Q6. Bonus

Create a piece of art (e.g. drawing, poem, short program, anything you like)! If you are unsure of what to make, select one member of the course staff and make art about that person. All exams will get full credit for this question even if you leave it blank.

Answer:

Extra Work Space

You may use this page for additional space for answers; keep it attached to this exam. Clearly note on the original question page that your answer is on this extra page, and clearly note on this page what question you are answering.

Answer: