

CS 220 - Spring 2025
Instructors: Mike Doescher and Louis Oliphant

Exam 1 — 10%

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

NetID (email): _____ @wisc.edu

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, fill in bubbles:
 - 001 - MWF 08:50 AM (Mike)
 - 002 - MWF 11:00 AM (Mike)
 - 003 - MWF 09:55 AM (Louis)
 - 004 - MWF 01:20 PM (Louis)
4. Under **F** of SPECIAL CODES, write *A* and fill in bubble **6**

If you miss step 4 above (or do it wrong), the system may not grade you against the correct answer key, and your grade will be no better than if you were to randomly guess on each question. So don't forget!

You may only reference your note sheet. You cannot use books, your neighbors, calculators, or other electronic devices during this exam. Please place your student ID face up on your desk. Turn off and put away portable electronics (including smart watches) now.

Use a #2 pencil to mark all answers. When you're done, please hand in the exam, note sheet, and filled-in scantron form. The note sheet will not be returned.

General

1. Consider the following code and its behavior:

```
def f(x, y):  
    if x:  
        if y:  
            return False  
        else:  
            return True  
    elif y:  
        return True  
    else:  
        return False
```

If `a1`, `a2` are Boolean values, which answer choice will always give the exact same output as `f(a1, a2)`?

- A. `a1 == a2`
 - B. `a1 != a2`
 - C. `a1 or a2`
 - D. `a1 or not a2`
 - E. `a1 and a2`
2. What is the output of the following code?

```
def func(x, y):  
    if x > 5:  
        if y < 6:  
            return x + y  
        else:  
            return x - y  
    else:  
        return x * y
```

`func(10, 2)`

- A. 8
- B. 20
- C. 12
- D. 5
- E. 6

3. Given the following code, what values are printed, and in what order?

```
i = 5
while True:
    print(i)
    if i == 1:
        break
    if i % 2 == 0:
        i = i // 2
    else:
        i = i * 3 + 1
```

- A. 5, 16, 8, 4, 2, 1
- B. 5, 16, 8.0, 4.0, 2.0, 1.0
- C. 5, 2, 7, 3, 1
- D. 5, 2.0, 7.0, 3.0, 1.0
- E. The code will result in an infinite loop of values being printed

4. Given the following code, what values are printed, and in what order?

```
i = 0
while i < 20:
    i += 1
    if i % 4 == 0:
        print(i)
```

- A. 0, 4, 8, 12, 16
- B. 4, 8, 12, 16
- C. 4, 8, 12, 16, 20
- D. 0, 4, 8, 12, 16, 20
- E. The code will result in an infinite loop of values being printed

-
5. Consider the following pseudocode and initial state of the variables:

Initial State:

$X = 5$

$Y = 3$

$Z = 2$

Pseudocode:

1. If X is greater than Y , subtract Z from X . Otherwise, add Z to Y .
2. If X is now odd, multiply X by Y . Otherwise, add Z to X .
3. If X is now greater than 10, set Y to $X - 4$. Otherwise, set Y to $X + 3$.
4. If Y is even, subtract 1 from Z . Otherwise, add 2 to Z .

What will be the final values of X , Y , and Z after executing the pseudocode?

- A. $X = 15$, $Y = 11$, $Z = 1$
- B. $X = 9$, $Y = 12$, $Z = 1$
- C. $X = 12$, $Y = 8$, $Z = 3$
- D. $X = 8$, $Y = 5$, $Z = 5$

-
6. Consider the following pseudocode and initial state of the variables:

Initial State:

```
age = 92
category = ""
flag = False
bonus = 0
```

Pseudocode:

1. If `age` is less than 18, set `category = "Too young"` and stop. Otherwise, continue to step 2.
2. If `age` is greater than 90, set `age = age - 10`, set `bonus = 5`, and continue to step 3. Otherwise, skip to step 5.
3. If `age` is divisible by 5, set `category = "Perfect age"` and go to step 6. Otherwise, continue to step 4.
4. If `age` is between 80 and 95 (inclusive; i.e., `80 <= age <= 95`), set `flag = True` and continue to step 5. Otherwise, go to step 6.
5. If `flag` is `True`, set `category = "Almost too old"` only if `category` is still an empty string (i.e., `category = ""`). Then, add `bonus` to `age` and go to step 6.
6. If `age` is greater than 100, set `category = "Too old"`. Otherwise, stop the execution.

What will be the final value of `category` after executing the pseudocode?

- A. "Perfect age"
- B. "Almost too old"
- C. ""
- D. "Too old"

7. What does the following code output?

```
print (3 ** (5 // 2) + 8 % 3 * 2 - 1)
```

- A. 7
- B. 9
- C. 10
- D. 12
- E. 13

8. What will be the final output of the following Python expression?

```
not ("A" * 2 == "AA" and "Hello" + "2" > "Hello1") or ("A" < "Z" and  
"3" * 2 != "333")
```

- A. True
- B. False
- C. `SyntaxError`
- D. "AAHello233"
- E. None of the above

9. Consider the following code snippet:

```
x = -5  
_y = 10  
1z = 15  
print(x + _y + 1z)
```

Which of the following best describes the issue in the code?

- A. The code will run successfully, and the output will be 20.
- B. The code will not run due to a Syntax Error in the variable name `_y`.
- C. The code contains a Semantic Error, as the expression does not produce the expected value.
- D. The code will result in a Runtime Error due to an invalid operation.
- E. The code will not run due to a Syntax Error in the variable name `1z`.

10. Consider the following code snippet:

```
a = 10  
b = 0  
result = (a > 5) or (b / 0 > 1)  
print(result)
```

What will be the output of this code?

- A. The code will output `True`.
- B. The code will output `False`.
- C. The code will output 0.
- D. The code will result in a Runtime Error due to division by zero.
- E. The code will not run due to a Syntax Error.

11. What will be the output of the following code snippet?

```
n = 2
m = 2048
a = 0
b = 5

while n <= m:
    n *= 2
    a += 1
    if a % 3 == 0:
        b += 2

print(a, b, n)
```

- A. 10 13 2048
- B. 11 13 2048
- C. 11 11 4096
- D. 10 11 4096
- E. None of the above

12. What will be the output of the following code snippet?

```
n = 5
m = 10
x = 15
y = 20
a = 0
while n <= m:
    n += 1
    a += 1
    while y > x:
        y -= 1
        a += 1

print(a)
```

- A. 30
- B. 25
- C. 11
- D. 10
- E. None of the above

13. You have access to a dataset of undergraduate Computer Science students at UWM through the `project` package, which provides the following functions:

- `project.count()`: Returns the number of records in the dataset as an **integer**.
- `project.get_major(idx)`: Returns the **string** representation of a student's major.
- `project.get_gpa(idx)`: Returns the **float** representation of a student's GPA. The maximum GPA of any student is 4.0.

The functions `get_major` and `get_gpa` take a row index (`idx`) as a parameter.

Now consider the following code snippet:

```
a = 5
b = ""

for idx in range(project.count()):
    gpa = project.get_gpa(idx)

    if gpa < a:
        a = gpa
        b = project.get_major(idx)
```

After executing this code, what does the variable `a` represent? You may assume that there isn't any invalid data in the dataset, meaning that the functions will not return `None` or empty string or negative numbers.

- A. The lowest GPA in the dataset
 - B. The highest GPA in the dataset
 - C. The major of the student with the lowest GPA
 - D. The major of the student with the highest GPA
 - E. The code will trigger an error
14. What will be the output of the following code snippet?

```
print(type(round(abs(float("-21.9233")), 2)))
```

- A. 21.92
- B. `<class 'float'>`
- C. `<class 'str'>`
- D. `<class 'int'>`
- E. -21.92

15. What will be the output of the following code snippet?

```
print("i", "^love", "cs", "220", sep = "^", end = "...")
```

- A. i ^love cs 220...
- B. i^lovecs220...
- C. i^^love^cs^220
- D. i^^love^cs^220...
- E. i love cs 220

16. Which of the following function calls will throw a `SyntaxError`?

```
def func(x, y=20, z=30):  
    return x + y + z
```

- A. `func(20, z=10, y=25)`
- B. `func(20, 10, 25)`
- C. `func(20, y=10, 25)`
- D. `func(20)`
- E. `func(20, 10, z=25)`

17. Consider the following code.

```
def a(g, h, i=30):  
    return (h + g - i)  
  
def b(x, y, z=20):  
    return a(x, y, z)  
  
def c(x, y, z=10):  
    return b(y, z, z=x)  
  
x = 5  
y = 2  
print(c(x, y))
```

What is the output of the line **print(c(x, y))**?

- A. 13
- B. 7
- C. -3
- D. 8
- E. 38

18. Consider the following code:

```
msg = "i"

def add_to_msg_helper(msg):
    msg = msg + "CS220"
    print(msg, end=" ")

def add_to_msg(msg):
    msg = msg + "Love"
    add_to_msg_helper(msg)
    return msg

print(msg, end=" ")
new_msg = add_to_msg(msg)
print(msg, end="")
```

What is the output after running the above code?

- A. i iLoveCS220 i
- B. i LoveCS220 i Love
- C. i LoveCS220 i
- D. i LoveCS220
- E. i CS220 i

19. What is the output after running the following code?

```
x = 10
y = 3

if y >= 6 and x < 11:
    print("A", end="")
elif y < 9 or x < 11:
    print("B", end="")
if y == "3" and x >= 3:
    print("C", end="")
if y > 4 or x < 14:
    print("B", end="")
else:
    print("C", end="")
```

- A. ABB
- B. BCB
- C. BCC
- D. BB
- E. AB

20. Consider the following code.

```
one = ??
two = ??

first = one and not two
second = (two or not one) and one

result = first and not second
```

What should the values of `one` and `two` be so that the variable `result` will be `True`?

- A. True, True
- B. True, False
- C. False, True
- D. False, False

Madison City Budget

For the following questions, you can assume that all member functions of the `project` module are correctly defined and will behave exactly as they did in P3.

The following functions are the same as in P3:

- `project.get_id(agency)`: Get the id of an agency.
- `project.get_budget(agency_id, year)`: Get the budget allocated to the agency represented by the id in that particular year.

21. Given the following function that calculates the average yearly change in value between two years, which of the following is the correct usage of the function if we want to calculate the average change of value between the year 2020 and 2024, given that the value associated with 2020 is 4 and the value associated with 2024 is 10?

```
def change_in_value_per_year(year1, value1, year2=2024, value2=6):  
    change_in_value = value2 - value1  
    num_years = year2 - year1  
    change_per_year = change_in_value / num_years  
    return change_per_year
```

- A. `change_in_value_per_year(year2 = 2024, 2020, value1 = 4, value2 = 10)`
- B. `change_in_value_per_year(year2 = 2024, year1 = 2020, value2 = 10, value1 = 4)`
- C. `change_in_value_per_year(2020, 2024, 10, 4)`
- D. `change_in_value_per_year(year1 = 2020, value1 = 4, 2024, 10)`

-
22. Given the following implementation that attempts to calculate the standard deviation of the budget allocated to the agency "Finance" from 2020 to 2024, which line of the code will trigger an error? Note: The backslash character (\) is used to explicitly break a long statement into multiple lines.

```
def std_agent(agency):  
    budget_2020 = project.get_budget(agency, 2020) #A  
    budget_2021 = project.get_budget(agency, 2021)  
    budget_2022 = project.get_budget(agency, 2022)  
    budget_2023 = project.get_budget(agency, 2023)  
    budget_2024 = project.get_budget(agency, 2024)  
    budget_avg = (budget_2020 +\  
                  budget_2021 +\  
                  budget_2022 +\  
                  budget_2023 +\  
                  budget_2024)/5 #B  
    std = (budget_2020 - budget_avg)**2 + \  
          (budget_2021 - budget_avg)**2 + \  
          (budget_2022 - budget_avg)**2 + \  
          (budget_2023 - budget_avg)**2 + \  
          (budget_2024 - budget_avg)**2 #C  
    std = (std / 5) ** 0.5 #D  
  
    print(std_agent("Finance"))
```

- A. A
- B. B
- C. C
- D. D
- E. The code is correct

Pokémon

Consider the following functions and table to answer the next several questions. `friend_status()` evaluates whether two Pokémon would be considered friends based on several pieces of information about them.

Assume that `get_region()`, `type1()`, `type2()`, `get_attack()`, `get_defense()`, `get_special_attack()`, `get_special_defense()`, `get_hp()`, `get_speed()` correspondingly return the region, type1, type2, attack, defense, special attack, special defense, HP, and speed of the Pokémon.

```
def get_stat_total(pkmn):
    stat_total = get_attack(pkmn) + get_defense(pkmn)
    stat_total += get_special_attack(pkmn) + get_special_defense(pkmn)
    stat_total += get_hp(pkmn) + get_speed(pkmn)
    return stat_total

def friend_status(pkmn1, pkmn2 = "Miraidon"):
    region1 = get_region(pkmn1)
    region2 = get_region(pkmn2)

    if region1 == region2:
        if type1(pkmn1) == type1(pkmn2) and type2(pkmn1) == type2(pkmn2):
            return "Best Friends"
        elif type1(pkmn1) == type1(pkmn2) or type2(pkmn1) == type2(pkmn2):
            return "Friends"
        else:
            return "Acquaintances"

    else:
        if type1(pkmn1) == type1(pkmn2) or type2(pkmn1) == type2(pkmn2):
            if abs(get_stat_total(pkmn1) - get_stat_total(pkmn2)) < 100:
                return "Friends"
            else:
                return "Acquaintances"

        elif abs(get_stat_total(pkmn1) - get_stat_total(pkmn2)) < 100:
            return "Acquaintances"
        else:
            return "Enemies"
```

The stats of the five Pokémon listed below will be used for the next three questions:

Name	Atk	Def	HP	Region	Sp Atk	Sp Def	Speed	Type 1	Type 2
Ivysaur	62	63	60	Kanto	80	80	60	Grass	Poison
Glimmora	55	90	83	Paldea	130	81	86	Rock	Poison
Weepinbell	90	50	65	Kanto	85	45	55	Grass	Poison
Coalossal	80	120	110	Galar	80	90	30	Rock	Fire
Miraidon	85	100	100	Paldea	135	115	135	Electric	Dragon

23. What is the output of `friend_status("Weepinbell", "Ivysaur")`?
- A. "Best Friends"
 - B. "Friends"
 - C. "Acquaintances"
 - D. "Enemies"
 - E. Syntax Error
24. What is the output of `friend_status("Coalossal")`?
- A. "Best Friends"
 - B. "Friends"
 - C. "Acquaintances"
 - D. "Enemies"
 - E. Syntax Error
25. What is the output of `friend_status(pkmn2 = "Miraidon", pkmn1 = "Glimmora")`?
- A. "Best Friends"
 - B. "Friends"
 - C. "Acquaintances"
 - D. "Enemies"
 - E. Syntax Error

-
26. Please select the code that, when used to replace the ..., correctly finishes implementing the `effective_damage` function shown below. Keep in mind that an attacker wants to do as much damage as possible, and if the attacker has two types, it can choose either type 1 or type 2 as its attack type. Assume `get_num_types()`, `type_bonus()` and `damage()` are implemented correctly and `type1()`, `type2()` get the type1 and type2 of a Pokemon respectively.

```
def effective_damage(attacker, defender):  
    if get_num_types(attacker) == 1:  
        return ( type_bonus(type1(attacker), defender)  
                * damage(attacker, defender) )  
    else:  
        bonus = ...  
        return bonus * damage(attacker, defender)
```

- A. `type_bonus(type1(attacker), defender) + type_bonus(type2(attacker), defender)`
- B. `max(type_bonus(type1(attacker), defender), type_bonus(type2(attacker), defender))`
- C. `type_bonus(type1(attacker), defender) * type_bonus(type2(attacker), defender)`
- D. `type_bonus(type1(attacker), defender) or type_bonus(type2(attacker), defender)`
- E. `min(type_bonus(type1(attacker), defender), type_bonus(type2(attacker), defender))`

Chess

Consider the following code for the questions in this section. This code attempts to draw an arbitrarily sized chess board containing one queen. The hashtag symbol (#) represents the black squares, spaces represent the white squares, and the Q represents the queen.

```
(1) def chess(x, y, rows=8, cols=8):
(2)     for r in range(rows):
(3)         for c in range(cols):
(4)             symbol = ""
(5)             if (r+c) % 2 == 0:      # print the black squares (#)
(6)                 symbol = "#"
(7)             elif (r+c) % 2 == 1:    # print the white squares
(8)                 symbol = " "
(9)             if (r == x and c == y): # print the Q
(10)                 symbol = "Q"
(11)             print(symbol, end="")
(12)         print()
```

The following function call generates the board below:

```
chess(3,3)
```

```
# # # #
# # # #
# # # #
# Q # #
# # # #
# # # #
# # # #
# # # #
```

27. How many times is the condition in line 9, `(r == x and c == y)`, evaluated when `chess(3,3)` is called?

- A. 0
- B. 1
- C. 8
- D. 32
- E. 64

28. If line 9 were changed to be an "else" (i.e. `else: # print the Q`), how many 'Q's would be printed for the same function call, `chess(3,3)`?

- A. 0
- B. 1
- C. 8
- D. 32
- E. 64

29. Which function call will print the following board?

```
# # #  
# #  
# # #  
# Q
```

- A. `chess(rows = 4, cols = 5)`
- B. `chess(3, 3, rows = 4, cols = 3)`
- C. `chess(3, 3, rows = 4, cols = 5)`
- D. `chess(4, 4, rows = 4, cols = 3)`
- E. `chess(4, 4, rows = 4, cols = 5)`

30. If the `%` in lines 5 and 7 were replaced with `//` how many `#` would be printed on the board when `chess(3,3)` is called?

- A. 1
- B. 2
- C. 3
- D. 31
- E. 32

Blank Page: this page is intentionally blank