

CS 220 - Spring 2023
Instructors: Mike Doescher and Gurmail Singh

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 01:20 PM (Gurmail)
 - 004 - MWF 03:30 PM (Gurmail)
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 and note sheet and your filled-in scantron form. The note sheet will not be returned.

Electric Vehicle Sales

1. The following function, `year_max(year)`, determines the highest number of sales in a given year. Choose the boolean expression that should replace `???`. Assume that the data is available only for years between 2017 and 2021 (inclusive).

```
def year_max(year):  
    if ???:  
        print("Invalid input for year")  
        return None  
    else:  
        # Perform sales of cars per year calculation
```

- A. `year <= 2016 and year >= 2022`
 - B. `year <= 2017 or year >= 2021`
 - C. `year < 2017 or year > 2021`
 - D. `year < 2017 and year > 2021`
 - E. `year > 2017 or year < 2021`
2. If we want to simplify the following function, `dip_in_sales(sales1, sales2, sales3)`, which expression can be used to replace the if statement blocks? Assume all parameters are initialized to integer values.

```
def dip_in_sales(sales1, sales2, sales3):  
    if sales1 > sales2:  
        if sales2 < sales3:  
            return True  
    return False
```

- A. `if sales1 < sales2 or sales2 > sales3`
- B. `if sales1 > sales2 or sales2 < sales3`
- C. `if sales1 > sales2 and sales2 < sales3`
- D. `if sales1 < sales2 and sales2 > sales3`
- E. None of the options implement equivalent logic

-
3. What value will be printed after running the following code snippet?

```
def mystery(y1, y2, s1, s2):  
    if y1 < y2:  
        if s1 < s2:  
            return "A"  
        else:  
            return "B"  
    if s1 < s2:  
        if y1 > y2:  
            return "C"  
        else:  
            return "D"  
    return "E"  
  
print(mystery(2021, 2017, 14000, 11000))
```

A. A B. B C. C D. D E. E

4. Assume that `project.get_sales("Tesla Model X", 2021)` returns an integer. What is the type of `x`?

```
sales = project.get_sales("Tesla Model X", 2021)  
y = str(sales)  
x = sales/5 + 500
```

A. int B. str C. float D. NoneType E. Runtime Error

Snake and Mouse

Reference the following code for the next 8 questions. The `draw` function below attempts to print a Snake and Mouse game ('S' represents positions occupied by the snake, 'M' represents the position of the mouse, and '.' represents empty space).

```
def draw (x, y, rows = 6, cols = 6):    #Line 1
    for i in range(0, rows):            #Line 2
        for j in range(0, cols):        #Line 3
            if i == x and j == y:       #Line 4 (print the M)
                print("M", end = "")    #Line 5
            elif i%4 == 0 or i%4 == 2:   #Line 6 (full row)
                print("S", end = "")    #Line 7
            elif i%4==1 and j==0:        #Line 8 (left S)
                print("S", end="")      #Line 9
            elif i%4==3 and j==cols-1:   #Line 10 (right S)
                print("S", end = "")    #Line 11
            else:                        #Line 12 (otherwise)
                print(".", end = "")    #Line 13
        print()                         #Line 14
```

Here is an example game:

```
SSSSSS
S.....
SSSSSS
...M.S
SSSSSS
S.....
```

5. Which of the following function calls will produce the example grid shown above?
- A. `draw(3, 3, 6)`
 - B. `draw(3, 6)`
 - C. `draw(4, 4)`
 - D. `draw(4, 4, 6, 6)`

-
6. The following print statement is equivalent to which of the following function calls?

```
print("SSS\n" + "SM.\n" + ("S") * 3)
```

- A. `draw(1, 1, 2, 3)`
 - B. `draw(1, 1, 3, 3)`
 - C. `draw(2, 2, 3, 3)`
 - D. `draw(2, 2, 3)`
7. Where would the function call `draw("0", "1")` print the "M"?
- A. "M" in the first row, second column
 - B. "M" in the top left corner
 - C. "M" in the second row, first column
 - D. "M" will not be printed on the board
8. Which of the following function calls will result in M not being printed?
- A. `draw(0,0)`
 - B. `draw(1,1)`
 - C. `draw(6,6)`
 - D. None of the above
9. How many M's will be printed if the `and` in Line 4 is changed to `or` when the function is called with the statement `draw(2, 2, 4, 4)`? NOTE: This change only applies to question.
- A. 4 B. 7 C. 8 D. 16
10. Which of the following is NOT a valid function call to `draw`?
- A. `draw(x = 1, y = 1)`
 - B. `draw(x=1, y=1)`
 - C. `draw(x = 1, y = 1, cols = 4, rows = 4)`
 - D. `draw(x = 1, rows = 4, cols = 4)`
 - E. `draw(1, 1, cols = 2)`
11. How many times is the condition in Line 8 evaluated when `draw(2, 2, 4, 4)` is called?
- A. 4 B. 8 C. 15 D. 16

General

12. Consider the following variables and their values.

`X = 7, Y = 1, Z = 0`

Evaluate the pseudocode:

1. If the value of X is greater than the value of Y, continue to step 2, otherwise skip to step 5
2. Subtract 2 from the value of X and store it in X
3. Add 1 to the value in Y and store it in Y
4. Go to step 1
5. Multiply the value of X and Y and store it in Z

What is the final value of Z after executing the above code?

- A. 0 B. 4 C. 7 D. 9 E. 10
13. Which of the following expressions does NOT evaluate to a value of type `float`?
- A. `55 // 2 + 2.0`
B. `-6.8`
C. `8 // 3`
D. `48 / 6`
E. `3 + 0.0`
14. What will the following expression evaluate to?

`3 * 3 ** 2 + 1 < 30 and not False or 5 != 7`

- A. `True`
B. `False`
C. `None`
D. `SyntaxError`

15. Consider the following code?

```
???                                #Line 1
squareroot_of_two = math.sqrt(2)  #Line 2
```

Which of the following should replace the ??? in line 1 to allow the code to run without error?

- A. `from math import *`
- B. `import math`
- C. `from math import sqrt as sqrt`
- D. `import math.sqrt`
- E. None of the above

16. Consider the following function:

```
def average_three(num1, num2 = 0, num3 = 0):
    return (num1 + num2 + num3) / 3
```

Which of the following function calls will result in a `SyntaxError`?

- A. `average_three(1, 1, 1)`
- B. `average_three(num2 = 1, num3 = 1, num1 = 1)`
- C. `average_three(1, num3 = 1)`
- D. `average_three(num1 = 1, 1, 1)`
- E. `average_three(1)`

17. Consider the following code

```
import math                                #Line 1
a = 5                                      #Line 2
b = 0                                      #Line 3
print("The quotient is ", a / b)          #Line 4
```

What type of error will be encountered upon running the following code snippet and what line is responsible for throwing this error?

- A. Syntax Error, Line 1
- B. Runtime Error, Line 4
- C. Syntax Error, Line 4
- D. Semantic Error, Line 2
- E. No Errors

18. Consider the following code snippet:

```
def add_nums(add1, add2):
    out = add1 + add2
    return out

def both_positive(num1, num2):
    comparison1 = add_nums(num1, num2) > num1
    comparison2 = add_nums(num1, num2) > num2
    return comparison1 and comparison2

x = 5
y = -3
both_positive(x,y)
```

Which of the following variables cannot be accessed in the scope of the function `both_positive()`?

- A. x
- B. num1
- C. comparison1
- D. out
- E. None of the above

19. What should you assign x and y, so “Green” is printed?

```
x = ???
y = ???

if (x or y) and (x and not y):
    print("Green")
else:
    print("Blue")
```

- A. x = False, y = False
- B. x = False, y = True
- C. x = True, y = False
- D. x = True, y = True

20. What are the values of price and total after the following code is executed?

```
weight = 1
num_items = 5
price = 0

if weight < 1:
    price = 5
if weight > 1:
    price = 15
else:
    price = 10
total = weight * price

if num_items > 5:
    total = total * 2
if num_items >= 5:
    total = total * 4
```

- A. price = 10, total = 20
- B. price = 15, total = 30
- C. price = 5, total = 20
- D. price = 10, total = 40
- E. price = 15, total = 60

21. Given the following code, which of the following letters will be printed and in what order?

```
x = 12

if x % 3 == 0:
    if x % 5 == 0:
        print("A")
    else:
        print("B")
if x % 3 == 0:
    print("C")
elif x % 4 == 0:
    print("D")
    if x % 1 == 0:
        print("E")
```

- A. A, C
- B. B, C, D
- C. B, C, D, E
- D. B, C

22. What are the values of a and b after the following code is executed?

```
a = -2
b = 2
while b != 0:
    if b > 0:
        b = b - 1
        a = a + 1
a = a + 1
```

- A. a = -2, b = 2
- B. a = 0, b = 0
- C. a = 0, b = 2
- D. a = 2, b = 0
- E. This code will create an infinite loop

23. How many times will “Hi” print out?

```
i = 0
j = 0
while i < 2:
    while j < 3:
        print("Hi")
        j += 1
    print("Hi")
    i += 1
```

A. 4 times B. 5 times C. 6 times D. 7 times E. 8 times

24. What will be printed out after running the following code?

```
def laugh_bot(i):
    laugh = ""
    while i >= 0:
        laugh = laugh + "Ha "
        i -= 1
    laugh = laugh + "!"
    return laugh
print(laugh_bot(4))
```

A. Ha Ha Ha Ha !
B. !
C. Ha Ha Ha Ha Ha !
D. HaHaHaHaHa!

Pokemon

Consider the following dataset and function for the following questions.

Name	Type 1	Type 2	Attack	Defense	HP	Legendary
Bulbasaur	Grass	Poison	100	85	90	False
Nidoking	Poison	Ground	102	77	78	False
Charizard	Fire	Flying	84	78	80	True
Articuno	Ice	Flying	85	100	90	True
Swinub	Ground	Ice	77	102	50	True

Assume that `type1(pkmn)`, `type2(pkmn)`, `attack(pkmn)`, `defense(pkmn)`, `hp(pkmn)`, and `is_legendary(pkmn)` correspondingly return the Type 1, Type 2, attack, defense, hp, and legendary status of each pokemon.

```
def battle(pkmn1 = "Charizard", pkmn2 = "Articuno"):
    if hp(pkmn1) > hp(pkmn2):
        if attack(pkmn1) > defense(pkmn2):
            print( pkmn1 + " attack first!")
            if is_legendary(pkmn1):
                return pkmn1 + " won legendarily!"
        elif attack(pkmn1) < defense(pkmn2):
            print( pkmn2 + " attack first!")
            if is_legendary(pkmn2):
                return pkmn2 + " won legendarily!"
    if type1(pkmn1) == "Fire" or type2(pkmn1) == "Fire":
        if type1(pkmn2) == "Ice" or type2(pkmn2) == "Ice":
            return "Incompatible. Can not battle!"
        else:
            return "It's a draw."
    else:
        return "Please choose another match."
```

25. What value will be **returned** by calling `battle()`?

- A. "Incompatible. Can not battle!"
- B. "Articuno won legendarily!"
- C. "Please choose another match."
- D. "It's a draw"
- E. "Articuno attack first!"

-
26. What value will be **returned** by calling `battle("Articuno", pkmn2="Swinub")`?
- A. "Articuno won legendarily!"
 - B. "Swinub won legendarily!"
 - C. "Please choose another match."
 - D. "Swinub attack first!"
 - E. `SyntaxError`
27. What value will be **returned** by calling `battle(pkmn1="Bulbasaur","Articuno")`?
- A. "Incompatible. Can not battle!"
 - B. "It's a draw."
 - C. "Please choose another match."
 - D. "Articuno won legendarily!"
 - E. `SyntaxError`
28. Suppose "Nidoking" is `pkmn1` and "Articuno" is `pkmn2`. In this case, what is the minimum number of argument(s) that must be passed to the function `battle`?
- A. 0 B. 1 C. 2 D. 3 E. 4

Consider the below function definition for the next two questions.

```
def strongest_pokemon(hp, type1 , type2 = "Flying", attack = 40):  
    if attack > 80:  
        if hp >= 50:  
            return True  
    elif not type1 == "Fire":  
        return False  
    if not type2 == "Flying":  
        return False  
    return True
```

29. Which function call will NOT return True?
- A. `strongest_pokemon(60, "Poison", attack = 81)`
 - B. `strongest_pokemon(hp = 60, type1 = "Poison", type2 = "Flying", attack = 81)`
 - C. `strongest_pokemon(attack = 81, type2 = "Flying", type1 = "Poison" , hp = 60)`
 - D. `strongest_pokemon(hp = 60, type1 = "Poison", attack = 81)`
 - E. `strongest_pokemon(hp = 60, type1 = "Poison", "Flying", 81)`
30. Which of the following lines of code can be used to shorten the `strongest_pokemon` function?
- A. `attack > 80 and (hp >= 50 or type1 == "Fire") and type2 == "Flying"`
 - B. `(attack > 80 and hp >= 50) or (not type1 == "Fire" and not type2 == "Flying")`
 - C. `(attack > 80 or hp >= 50) and (type1 == "Fire" or type2 == "Flying")`
 - D. `(attack > 80 and hp >= 50) or (type1 == "Fire" and type2 == "Flying")`
 - E. `attack > 80 or (hp >= 50 and type1 == "Fire") or type2 == "Flying"`

(Blank Page: you may tear this off for scratch work, but hand it in with your exam)

(Blank Page: you may tear this off for scratch work, but hand it in with your exam)