

CS 220 - Fall 2021
Instructors: Meenakshi Syamkumar, Andrew Kuemmel

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 11:00am (Meena)
 - 002 - MWF 1:20pm (Meena)
 - 003 - MWF 8:50am (Andy)
 - 004 - MWF 9:55am (Andy)
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!

Many of the problems in this exam are related to the course projects, but some questions assume the availability of slightly different functions (e.g., for accessing the data). We won't have any trick questions where we call a function that doesn't exist and you need to notice. Thus, if you see a call to a function we haven't explicitly defined in the problem, assume the function was properly implemented (perhaps immediately before the code snippet we DO show) and is available to you.

You may only reference your notesheet. You may not use books, your neighbors, calculators, or other electronic devices on 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 these sheets in addition to your filled-in scantron.

(Blank Page)

General

1. Which of the following expressions evaluate to type **int**?

A. `2 + 2.0` B. `20 / 2` C. `"22" + "0"` **D. `20 // 2`** E. `9 ** (1 / 2)`

2. What will be printed by executing the following sequence of shell commands?

```
echo one > A.txt
echo two > B.txt
echo three > C.txt
cat B.txt > C.txt
mv C.txt D.txt
cp D.txt A.txt
cat A.txt
```

A. one **B. two** C. three D. A.txt E. none of the above due to an error

3. Which expression evaluates to 24?

A. `2 * 4 + 8` B. `2 ^ 3 * 3` **C. `2 ** 3 * 3`** D. `2 + 2 * 2 * 3` E. `26 - - 2`

4. A function without a return statement will automatically return None.

A. True B. False

5. Which statement is **true** about functions?

A. named keyword arguments can be passed before positional arguments
B. multiple return statements may execute in a single function invocation
C. multiple return statements cannot be written in a single function definition
D. passing arguments to parameters with default arguments is optional
E. the same function cannot be invoked multiple times in a program

6. What type will be printed?

```
course = 220
print(type("CS" + course))
```

A. string B. float C. bool D. int **E. none of the above due to an error**

7. Which of the following is a valid variable name?

A. `"odd numbers"` **B. `_odd_numbers_`** C. `odd|even` D. `number%2` E. `1_mod`

8. What is the output of following code?

```
a = not False and not True or False
b = 4 * 2 ** 2 // 3
print(a, b)
```

A. True5 B. False 4 C. False 21 D. True 5 **E. False 5**

9. What does the following code print?

```
def foo():
    x = 2
    return 20
x = foo()
print(x)
```

A. 2 **B. 20** C. 220 D. None E. none of the above due to an error

10. What does the following code print?

```
i = 1
s = ""
while i <= 5:
    i += 1
    if i % 2 != 0:
        s += str(i)
    else:
        continue
    s += ";"
print(s)
```

A. 35 B. 1 C. 3;5; D. 2;4 E. 1;3;5

11. What kind of error is present in the following code?

```
def power(base, exponent = 2):
    return base * exponent
some_num = 10
some_num_squared = power(some_num)
print(some_num_squared)
```

A. Syntax B. Runtime **C. Semantic** D. No error

12. What is the output of the following code?

```
n1 = 4
n2 = 5
sign = "+"

def calculator(n1, n2, sign = "multiply"):
    if sign == "multiply":
        return n1 * n2
    elif sign == "minus":
        return n1 - n2
    elif sign == "divide":
        return n1 / n2
    elif sign == "plus":
        return n1 + n2
    else:
        return "Invalid sign!"

print(calculator(n1, 10, sign))
```

A. 40 B. 9 C. 14 D. 20 **E. Invalid sign!**

13. What is printed?

```
def c(x = 0, y = 0):
    print(x, y)

def b(x, y):
    y = 1
    c(y + 1)

def a(x, y):
    b(y, y = x)

x = 1
y = 2
a(y, x)
```

A. 0 0 B. 3 0 C. 1 2 **D. 2 0** E. 0 3

14. Which of these will exactly print the text below?

The dog says "woof woof"

- A. `print("The dog says "woof woof")`
- B. `print("The dog says", end = " \"woof woof\"\\n")`
- C. `print("The dog", "woof woof", sep = "says")`
- D. `print("The dog says \"woof woof\"")`
- E. **Options B and D are both correct**

15. If you want to stop all iterations of a loop, you should use:

- A. `continue` B. `break` C. `stop` D. `pass`

16. What is printed?

```
text = "aBCd4"
final = ""
for c in text:
    if c.islower():
        final += c.upper()
print(final)
```

- A. **AD** B. BC C. ABCD D. ABCD4 E. none of the above due to an error

17. Determine the outcome of the below pseudocode.

State:

X

Y

Code:

1. Add the value in X to the value in Y and store the result in Y.
2. Add the value in Y to the value in X and store the result in X.

- A. X = 30, Y = 40 B. X = 40, Y = 40 C. **X = 40, Y = 30** D. X = 10, Y = 30

**Blank Page: this page is intentionally blank,
please turn over for next section questions**

Pokémon

Consider the following table and function to answer the next several questions. The function evaluates whether two Pokémon would be considered friends based on a variety of criteria.

Name	Region	Type 1	Type 2	Speed
Butterfree	Kanto	Bug	Flying	70
Squirtle	Kanto	Water	Normal	43
Omanyte	Kanto	Rock	Water	35
Sandile	Unova	Ground	Rock	65
Krokorok	Unova	Ground	Dark	74

Assume that `region(pkmn)`, `type1(pkmn)`, `type2(pkmn)`, `speed(pkmn)` correspondingly return region, type1, type2, speed of the Pokémon.

```
def friend_status(pkmn1, pkmn2 = "Butterfree"):
    region1 = region(pkmn1)
    region2 = region(pkmn2)
    if region1 == region2:
        if type1(pkmn1) == type1(pkmn2):
            if abs(speed(pkmn1) - speed(pkmn2)) <= 20:
                return "Best friends"
        elif type2(pkmn1) == type2(pkmn2):
            if abs(speed(pkmn1) - speed(pkmn2)) <= 20:
                return "Good friends"
        elif type1(pkmn1) == type2(pkmn2) or type2(pkmn1) == type1(pkmn2):
            if abs(speed(pkmn1) - speed(pkmn2)) <= 20:
                return "Acquaintances"
    if type1(pkmn2) == "Flying" or type2(pkmn2) == "Flying":
        if type1(pkmn1) == "Rock" or type2(pkmn1) == "Rock":
            return "Enemies"
    else:
        return "Acquaintances"
    return "Enemies"
```

18. What does `friend_status("Squirtle", "Omanyte")` evaluate to?
- A. Best friends
 - B. Good friends
 - C. Acquaintances**
 - D. Enemies
 - E. Syntax error

-
19. What does `friend_status("Sandile")` evaluate to?
A. Best friends B. Good friends C. Acquaintances **D. Enemies** E. Syntax error
20. What does `friend_status(pkmn1 = "Butterfree", "Squirtle")` evaluate to?
A. Best friends B. Good friends C. Acquaintances D. Enemies **E. Syntax error**
21. What does the following code print?
- ```
pkmn = "Butterfree"
print(pkmn[6:9] + pkmn[9])
```
- A. ree   B. rfre   **C. free**   D. freee   E. Index error

Consider the below function definition for the next two questions.

```
def evaluate(attack, spl_attack = 30, defense = 50, hp = 50):
 if attack <= 50:
 if spl_attack == 30:
 if defense > 40:
 if hp != 80:
 return True
 return False
```

22. How many parameters have default arguments in `evaluate` function?  
A. 1   B. 2   **C. 3**   D. 4
23. Which of the following lines of code can be used to shorten the `evaluate` function definition?
- A. `return not(attack > 50 or spl_attack != 30 or defense <= 40 or hp == 80)`
  - B. `return not(attack > 50 and spl_attack != 30 and defense <= 40 and hp == 80)`
  - C. `return not(attack <= 50 and spl_attack == 30 and defense > 40 and hp != 80)`
  - D. `return not(attack <= 50 or spl_attack == 30 or defense > 40 or hp != 80)`

---

## Energy Consumption

24. Assume `get_idx(source)` returns the correct index for a given energy source and `get_consumption(index, year)` returns the correct consumption in a specific year, given an index. For which energy source does the following code print the consumption?

```
biomass_energy = "Solar Energy"
hydroelectric_energy = biomass_energy
source_idx = get_idx(hydroelectric_energy)
consumption = get_consumption(source_idx, 2020)
biomass_energy = "Wood Energy"
print(consumption)
```

- A. Conventional Hydroelectric Power
  - B. Solar Energy**
  - C. Wind Energy
  - D. Wood Energy
  - E. Biomass Energy
25. What should replace `????` if only years 2015 through 2020 (inclusive) are allowed as arguments to this function?

```
def change_per_year(energy, start_year, end_year):
 if start_year >= end_year:
 print("Invalid input for start year!")
 elif ????:
 # code returns change when start_year and end_year are in range
 else:
 print("Invalid input for year!")
 # implicit return None
```

- A. `(start_year <= 2015 and start_year >= 2020) and (end_year <= 2015 and end_year >= 2020)`
- B. `(start_year > 2015 or start_year < 2020) or (end_year > 2015 or end_year < 2020)`
- C. `(start_year > 2015 or start_year < 2020) and (end_year > 2015 or end_year < 2020)`
- D. `(start_year >= 2015 and start_year <= 2020) or (end_year >= 2015 and end_year <= 2020)`
- E. `(start_year >= 2015 and start_year <= 2020) and (end_year >= 2015 and end_year <= 2020)`**

- 
26. Which kind of import at ??? would not allow us to call the `ceil` function as we do on the next line?

```
???
ceil_val = ceil(10.45)
```

A. `import math`   B. `from math import *`   C. `from math import ceil`

27. Assuming the following, what gets printed?

```
get_consumption("Wood Energy", 2018) returns 10.5
get_consumption("Wood Energy", 2019) returns 8.4

def change(y1, y2):
 c1 = get_consumption("Wood Energy", y1)
 c2 = get_consumption("Wood Energy", y2)

 if y2 < y1:
 if c2 < c1:
 return "A"
 else:
 return "B"
 else:
 if c1 < c2:
 return "C"
 else:
 return "D"
 return "E"

print(change(2019, 2018))
```

A. A   B. B   C. C   D. D   E. E

---

28. What will be printed?

```
def fun(year1, year2):
 # Assume change_rate1 is 0.5 (change_per_year function call return value)
 # Assume change_rate2 is -0.5 (change_per_year function call return value)
 consumption_target = 0

 if change_rate1 > change_rate2:
 consumption_target += 50
 elif change_rate2 > change_rate1:
 consumption_target += 50
 else:
 consumption_target += 50
 if year1 > year2:
 consumption_target -= 25
 elif year2 >= year1:
 consumption_target -= 25
 return consumption_target

print(fun(2016, 2020))
```

A. -25   B. 0   C. 25   D. 50   E. 100

29. Which of the following options would simplify the body of the `will_overtake` function? Assume that all arguments are float values.

```
def will_overtake(change_rate1, energy1, change_rate2, energy2):
 # energy1, energy2 parameters will be initialized to 2020 consumption
 if change_rate1 > change_rate2:
 if energy2 >= energy1:
 return True
 if change_rate2 > change_rate1:
 if energy1 >= energy2:
 return True
 return False
```

- A. `return (change_rate1 > change_rate2 and energy2 >= energy1) == (change_rate2 > change_rate1 and energy1 >= energy2)`
- B. `return (change_rate1 > change_rate2 or energy2 >= energy1) and (change_rate2 > change_rate1 or energy1 >= energy2)`
- C. `return (change_rate1 > change_rate2 and energy2 >= energy1) or (change_rate2 > change_rate1 and energy1 >= energy2)`
- D. `return (change_rate1 > change_rate2 and energy2 >= energy1) and (change_rate2 > change_rate1 and energy1 >= energy2)`

- 
30. What should replace ???? to compute the average “Wind Energy” consumption over the range of years between start and end?

```
def avg_consumption(start_year = 2015, end_year = 2020):
 total = 0

 year = start_year
 while year <= end_year:
 # assume returns energy consumption in given year
 total += get_consumption("Wind Energy", year)
 year += 1
 avg = total / ????

 return avg
```

- A. 6
  - B. 2020
  - C. end\_year
  - D. (end\_year - start\_year + 1)
  - E. (end\_year - start\_year)
31. What will be the type of avg in the previous question?
- A. int   B. str   C. bool   D. list   **E. float**

---

# Tic-Tac-Toe

Reference the following code for the next few questions. The code attempts to draw a tic-tac-toe board.

```
X| |
-+-+--
| |
-+-+--
| |
```

```
def draw(x = 0, y = 0, move = "X"): # line 1
 i = 0 # line 2
 while(i < 5): # line 3
 if i % 2 != 0: # line 4
 print("-+-+--", end = "") # line 5
 else: # line 6
 j = 0 # line 7
 while j < 5: # line 8
 if j % 2 != 0: # line 9
 print("|", end = "") # line 10
 elif i == 2 * x and j == 2 * y: # line 11
 print(move, end = "") # line 12
 else: # line 13
 print(" ", end = "") # line 14
 j += 1 # line 15
 print() # line 16
 i += 1 # line 17
```

32. What does `draw(3, "X", y = 3)` evaluate to?
- A. X in the bottom-right corner of the board
  - B. X in the top-left corner of the board
  - C. Empty tic-tac-toe board
  - D. X in the bottom-left corner of the board
  - E. Syntax error**

---

33. How can we modify the while loop on line 3 to create the below 2 x 3 grid?

```
X| |
-+-+--
| |
```

- A. `while(i < 1):`
- B. `while(i < 2):`
- C. `while(i < 3):`
- D. `while(i < 4):`
- E. `while(i <= 4):`

34. What does `draw(2, 2, "O")` evaluate to?

- A. O in the bottom-right corner of the board**
- B. X in the top-left corner of the board
- C. X in the bottom-right corner of the board
- D. O in the bottom-left corner of the board
- E. Syntax error

35. Which of the following lines of code, when commented out, will cause an infinite loop?

- A. line 3   B. line 7   C. line 8   **D. line 15**   E. line 16

---

(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)