

CMPT 120-D400 Midterm Exam 1, Fall 2022

This is a **90 minute closed book exam**: notes, books, computers, calculators, electronic devices, etc. are **not** permitted. Do not speak to any other students during their exam or look at their work. Please remain seated and **raise your hand** if you have a question.

Sums and Characters

a) (5 marks) The sum of the integers from a to b is given by this formula:

$$\frac{(b - a + 1)(a + b)}{2}$$

Write a Python function called `int_sum(a, b)` that uses this formula (and no loops!) to calculate and return the sum of the integers from a to b . You can assume a and b are both integers, and a is less than, or equal to, b .

Make sure `int_sum` returns its result as an `int`, *not* a `float`.

Here are some examples of how it should work:

```
>>> int_sum(4, 7)
22
>>> int_sum(1, 3)
6
>>> int_sum(-100, 100)
0
```

Possible Solutions

```
def int_sum(a, b):
    return (b - a + 1) * (a + b) // 2
```

```
def int_sum(a, b):
    return int((b - a + 1) * (a + b) / 2)
```

```
def int_sum(a, b):
    return int(0.5 * (b - a + 1) * (a + b))
```

b) (5 marks) Write a Python function called `describe(s)` that returns a string describing the length of string `s`. It should work *exactly* like this:

```
>>> describe('M')           # strings of length 1 have a special format
'M is a single character'

>>> describe('wigs')
'cherry is 4 characters long'

>>> describe('cherry')
'cherry is 6 characters long'

>>> describe(' ')
' is 0 characters long'      # only a space before "is"
```

Possible Solutions

```
def describe(s):
    n = len(s)
    if n == 1:
        return s + ' is a single character'
    else:
        return s + ' is ' + str(n) + ' characters long'
```

```
def describe(s):
    if len(s) == 1:
        return s + ' is a single character'
    else:
        return s + ' is ' + str(len(s)) + ' characters long'
```

```
def describe(s):
    n = len(s)
    if n == 1:
        return f'{s} is a single character'
    else:
        return f'{s} is {n} characters long'
```

Arcade Bowling

(5 marks) An arcade's mini bowling game gives you tickets depending upon your final score:

Score	Tickets
0	1
1 to 5	3
6 to 11	5
12 or higher	10

A score of less than 0 means the machine's cheat-detection sensor has been triggered, and the number of tickets is 0.

Write a Python function called `tickets(score)` that takes an integer `score` as input, and returns the number of tickets for that score. You can assume `score` is a valid Python `int`.

```
>>> tickets(0)
1
>>> tickets(4)
3
>>> tickets(11)
5
>>> tickets(20)
10
>>> tickets(-3)
0
```

Make your function reasonably efficient: it **shouldn't** do any unnecessary computation.

Possible Solution

```
def tickets(score):  
    if score < 0:  
        return 0  
    elif score == 0:  
        return 1  
    elif 1 <= score <= 5:  
        return 3  
    elif 6 <= score <= 11:  
        return 5  
    else:  
        return 10
```

Loops

a) (5 marks) Write a complete Python program that uses a **while-loop** (and no other kind of loop) to print the numbers from 5000 down to 1, and “Blast off!” on the last line:

```
5000
4999
4998
...
3
2
1
Blast off!
```

Possible Solution

```
n = 5000
while n > 0:
    print(n)
    n -= 1
print('Blast off!')
```

b) (5 marks) Write a complete program that asks the user to enter the string “yes” or “no”. If they **don’t** enter “yes” or “no” then it asks them to try again (as shown in the sample run). It keeps re-asking them until they enter “yes” or “no”, and finally prints “Got it!” when they do

Here are three sample runs:

```
Yes or no? yep  
Sorry, I don't understand that.
```

```
Yes or no? nope  
Sorry, I don't understand that.
```

```
Yes or no? yes  
Got it!
```

Possible Solution

```
ans = input('Yes or no? ') # get input: 1 mark  
while not (ans == 'yes' or ans == 'no'): # header/condition: 2.5 marks  
    print('Sorry, I don\'t understand that.') # body: 1 mark  
    ans = input('Yes or no? ')  
print('Got it!') # final message: 0.5 marks
```

Multiple Choice

For each of the following questions, fill in **the one best answer** on the answer sheet.

Every correct answer is worth 1 mark. Incorrect answers, unanswered questions, questions with more than one answer, or questions with illegible answers, are worth 0.

1) Consider these two statements:

- i) `3 + "a"` evaluates to the string `3a`
- ii) `3 * "a"` evaluates to the string `aaa`

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

2) Compared to popular programming languages like C++ or Java, Python programs tend to:

- A. use much less memory
- B. run much more quickly
- C. run much more slowly
- D. take longer for the programmer to create

3) What is **source code**?

- A. program instructions, typically appearing in a text file
- B. the output of an interpreter or compiler
- C. the idea of the program the programmer has in their mind
- D. it's another name for pseudocode

4) Consider these two statements:

- i) `print("5" + "3")` and `print("5 + 3")` both print exactly the same thing
- ii) `print(5 + 3)` and `print(3 + 5)` both print exactly the same thing

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false


```
name = input('Who are you? ') # line 1
print('Hello ' + name + '!')  # line 2
```

Program Listing 1

- 5) This question is about **Program Listing 1**.

The program has exactly 2 statements.

- A. True
B. False

- 6) This question is about **Program Listing 1**.

The program has exactly 2 different source code comments.

- A. True
B. False

- 7) This question is about **Program Listing 1**.

The program will work the same if line 2 is replaced by this:

```
print(f'Hello {name}!')
```

- A. True
B. False

- 8) This question is about **Program Listing 1**.

The = in the line 1 is Python's equality operator.

- A. True
B. False

- 9) This question is about **Program Listing 1**.

The program has a syntax error.

- A. True
B. False

10) It's possible for a single Python value to be both of type `float` and type `str`.

- A. `true`
- B. `false`

11) What is the type of this value?

`39.0`

- A. `int`
- B. `float`
- C. either `int` or `float`, depending on how it is used
- D. `num`

12) Consider these two statements:

- i) In Python, a variable name can contain the `?` character (question mark).
- ii) In Python, a variable name can contain the `_` character (underscore).

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

13) How many of these are Python operators?

`%`
`%%`
`*`
`**`
`/`
`//`

- A. 2
- B. 3
- C. 4
- D. 5
- E. 6

14) Consider these two statements:

- i) Python's + operator has higher precedence than its * operator.
- ii) Python's * operator has higher precedence than its / operator.

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

15) Python's math module has a variable called pi with the value of 3.141592653589793.
What does this program print?

```
import math
```

```
pi = -75  
print(math.pi)
```

- A. 3.141592653589793
- B. -75
- C. nothing: the program has an error and doesn't run correctly

16) Consider these two statements:

- i) print is one of Python's standard modules.
- ii) turtle is one of Python's standard modules.

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

- 17) Suppose the Python file named `days.py` contains various functions for dealing with days of the week.

Suppose the Python program file `calendar.py` is in the same folder as `days.py`. How can you import the code in `days.py` into `calendar.py`?

- A. `import days`
- B. `import days.py`
- C. `import calendar`
- D. `import calendar.py`

- 18) In Python, a **boolean** expression is any expression that:

- A. has either True or False somewhere in it
- B. can evaluate to exactly two different possible values
- C. evaluates to either True or False

- 19) Here is a truth table that defines how some boolean operator **op** works:

a	b	a op b
False	False	False
False	True	True
True	False	True
True	True	False

What Python operator is **op**?

- A. and
- B. or
- C. not
- D. none of the above

- 20) Suppose `s` and `t` are *different* Python strings.

Consider these two statements:

- i) `s != t` always evaluates to False.
- ii) `s <= t` always evaluates to False.

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

21) Consider these two statements:

- i) If the Python expression `a and b` is True, then the expression `a or b` must be True.
- ii) If the Python expression `a and b` is False, then the expression `a or b` must be False.

Which one of these statements most accurately describes the truth values of i) and ii)?

- A. i) and ii) are both true
- B. i) and ii) are both false
- C. i) is false and ii) is true
- D. i) is true and ii) is false

22) What does this print?

```
n = 4
if n <= 4:
    print('up')
elif n == 4:
    print('down')
elif n > 3:
    print('left')
else:
    print('none')
```

- A. up
- B. up
down
- C. up
down
left
- D. none