

First Midterm Exam  
CSCI 1100 Gateway to Computer Science

KEY

Thursday October 13, 2022  
Instructor Muller  
Boston College  
Fall 2022

Before reading further, please arrange to have an empty seat on either side of you.

Now that you are seated, please write your name **on the back** of this exam.

This is a closed-book exam. Computers, calculators, and books are prohibited. In solving problems involving repetition, you are free to use any form that you would like. Partial credit will be given so be sure to show your work. **Please try to write neatly.**

Problem	Points	Out Of
1		3
2		2
3		2
4		3
5		3
6		3
7		4
Total		20

1. (3 Points) For each of the following, indicate what would happen if the code was evaluated in a Python shell. If the code would produce a value, what value would it produce? If the code would produce an error, what error?

(a) 

```
def f(x):  
    return (g(x), g(y))  
  
def g(y): return y * 2  
  
f(4)
```

**Answer:**

This is an error, y isn't bound in f.

(b) 

```
def senior(age):  
    return age >= 65  
  
"Alice" if senior(45) else "Mei"
```

**Answer:**

"Mei"

(c) 

```
def puzzle(m, n):  
    return [ [ 5 for i in range(m) ] for j in range(n) ]  
  
puzzle(2, 3)
```

**Answer:**

[[5, 5], [5, 5], [5, 5]]

2. (2 Points) *Compounded growth* of a quantity is usually expressed as a periodic percentage together with the number of periods of growth. In particular, a *present value*  $p_v$  of a quantity growing at  $i$  percent per period over  $n$  compounding periods will grow to a *future value*

$$f_v = p_v(1 + i)^n.$$

The flip side of this can be found by dividing both sides by the compounding factor  $(1 + i)^n$ . Given a future value, the growth rate and the number of compounding periods, the present value is  $p_v = f_v / (1 + i)^n$ .

Write a function `presentValue : float * float * int -> float` so that `presentValue(fv, i, n)` computes the present value of  $f_v$  over  $n$  periods.

**Answer:**

```
# presentValue : float * float * int -> float
def presentValue(fv, i, n):
    return fv / (1.0 + i) ** n
```

3. (2 Points) Young folks and old folks get a discount when attending the local fair. People 18 or under or 65 or older pay \$1, The rest of us pay 5 bucks. Write a function `ticketPrice : age -> int` that computes the price of a ticket.

**Answer:**

```
# ticketPrice : int -> int
def ticketPrice(age):
    return 1 if age <= 18 or age >= 65 else 5

def ticketPrice(age):
    if age <= 18 or age >= 65:
        return 1
    else:
        return 5
```

4. (3 Points) Let's say we have a 3-sided die with labels A, B and C. The die is weighted so that face A comes up half the time while B and C each come up 25% of the time. Write a function `threeSide : unit -> string` that returns the string "A", "B" or "C" with the likelihood described above.

Answer:

```
# threeSide : unit -> string
def threeSide():
    r = random.random()
    if r < .5:
        return "A"
    else:
        if r < .75:
            return "B"
        else:
            return "C"
```

5. (3 Points) Write a Python function `copy : item * int -> list item` such that a call `copy(item, n)` produces a list of `n` copies of `item`. For example, the call `copy("Go", 3)` should return the list `["Go", "Go", "Go"]`. The call `copy("Alice", 0)` should evaluate to the empty list `[]`.

Answer:

```
# copy : item * int -> list item
def copy(item, n):
    return [ item for _ in range(n) ]
```

6. (3 Points) Let's say we have a list of student information of the following form.

```
example = [("Alice", 2026, 19), ("Mei", 2025, 20), ... ]
```

Each 3-tuple in the list represents a student's name, graduation year and age. Write a function `averageAge : students -> float` such that a call `averageAge(students)` returns the average age of the students.

**Answer:**

```
# averageAge : list students -> float
def averageAge(students):
    sum = 0
    for (_, _, age) in students:
        sum = sum + age
    return sum / len(students)
```

7. (4 Points) A *composite number* is a positive integer that can be formed by multiplying two smaller positive integers. Write the function `isComposite : int -> bool` such that a call `isComposite(n)` returns `True` if `n` is composite. Otherwise `isComposite` should return `False`.

**Answer:**

```
# isComposite : int -> bool
def isComposite(n):
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return True
    return False
```