

# CS 111 - Winter 2020

## Quiz 2 Practice Questions

The quiz will be worth 30 points, and each point should correspond roughly to one minute of your time. This list of practice questions is not necessarily representative of the length of the quiz – it is hopefully longer to give you more practice, but there might be more questions in a given category on the actual quiz.

### True/False questions (approximately 2 points)

- 1) Every Python function returns some value.
- 2) Information can be passed into a function through parameters.
- 3) Variables defined in a function are local to that function.
- 4) Python functions can only modify a parameter if it is mutable.
- 5) The Boolean operator `or` returns `True` when both of its operands are true.
- 6) `a and (b or c) == (a and b) or (a and c)`
- 7) `True or False`

### Multiple-choice questions (approximately 3 points)

- 1) What expression would create a line from (2,3) to (4,5)?
  - (a) `Line(2, 3, 4, 5)`
  - (b) `Line((2,3), (4,5))`
  - (c) `Line(2, 4, 3, 5)`
  - (d) `Line(Point(2,3), Point(4,5))`
- 2) A Python function definition begins with
  - (a) `def`
  - (b) `define`
  - (c) `function`
  - (d) `func`
- 3) Which of the following is *not* a step in the function-calling process?
  - (a) The calling program suspends.
  - (b) The format parameters are assigned the value of the actual parameters.
  - (c) The body of the function executes.
  - (d) Control returns to the point just before the function was called.
- 4) A function with no `return` statement returns
  - (a) nothing
  - (b) its parameters
  - (c) its variables
  - (d) `None`

## Code prediction (approximately 9 points)

1) Sketch the shape that would be drawn by the following code. Make sure to label a few coordinates, but you don't need to draw it perfectly to scale.

```
win = GraphWin("My window", 400, 200)
rect = Rectangle(Point(20, 20), Point(300, 100))
rect.draw(win)
```

2) Predict the output that would be generated by the following code.

```
a = 4
b = 10
c = 22

if a > b:
    if b < c:
        print("Apple")
    else:
        print("Banana")
elif b > c:
    if a == b:
        print("Cat")
    elif a < b:
        print("Dog")
else:
    print("Elephant")
    if a < c:
        print("Fish")
    if a < b:
        print("Giraffe")
    else:
        print("Hotel")
print("Igloo")
```

3) Write a truth table that shows the Boolean value of each of the following Boolean expressions.

(a) not (P and Q)

(b) (not P) and Q

(c) (not P) or (not Q)

4) Predict the output that would be generated by the following program.

```
def myFunction(x, y):
    return x * 2 + y

def main():
    a = 12
    b = 3
    c = myFunction(a, b)
    d = myFunction(1, c)

    print(c)
    print(d)

if __name__ == "__main__":
    main()
```

5) Predict the output that would be generated by the following code.

```
x = 10
while x > 0:
    x = x - 3
    print(x)
```

6) Predict the output that would be generated by the following code.

```
x = 1
y = 12
while x <= y:
    if y % x == 0:
        print(x)
    x = x + 1
```

### Rewriting code (approximately 6 points)

1) Rewrite the following for loop to be a while loop instead.

```
mylist = [1,2,3,4]
for i in range(len(mylist)):
    print("Element", i, "is", mylist[i])
```

2) The Boolean expressions in the following if and elif conditions are overly complex. Use truth tables or logic to replace them with simpler expressions.

```
if (a and b) or (a and not b):
    print("something")
elif (a and False) or (True or not b):
    print("some other thing")
```

## Fixing bugs (approximately 4 points)

1) A person is eligible to be a US senator if they are at least 30 years old and have been a US citizen for at least 9 years. To be a US representative these numbers are 25 and 7, respectively. The following program accepts a person's age and years of citizenship as input and attempts to output their eligibility for the Senate and House. Unfortunately, there are three bugs in the code. What are they?

```
def checkSenateEligibility(age, years):
    if age >= 30:
        return True
    if years >= 9:
        return True
    else:
        return False

def checkHouseEligibility(years, age):
    if age < 25:
        return False
    elif years < 7:
        return False

def main():
    age = int(input("What is your age? "))
    years = int(input("For how many years have you been a US citizen? "))

    isEligibleForSenate = checkSenateEligibility(age, years)
    isEligibleForHouse = checkHouseEligibility(years, age)

    print("You are eligible for the Senate:", isEligibleForSenate)
    print("You are eligible for the House:", isEligibleForHouse)

if __name__ == "__main__":
    main()
```

## Writing code (approximately 6 points)

1) Write a program that draws two dice next to each other in a window (so really, just two squares). The values should be one, two, or three, depending on two random numbers. For one, the circle on the die should be centered. For two, they should be approximately  $1/4$  and  $3/4$  along the bottom-left to upper-right diagonal. For three, they should be approximately  $1/4$ ,  $1/2$ , and  $3/4$  along the bottom-left to upper-right diagonal.

2) Many companies pay time-and-a-half for any hours worked above 40 in a given week. Write a program to input the number of hours worked and the hourly rate and calculate the total wages for the week.

3) Write a function that takes a value  $n$  and returns the number of times that  $n$  can be divided by 2 (using integer division) before reaching 1. Your function should use a `while` loop.

```
def numberTimesDivisible(n):
    """
    Returns the number of times n can be divided by 2
    (integer division) before reaching 1.
    Assumes n is at least 1.

    ex: n=1 -> 0
        n=2 -> 1
        n=3 -> 1
        n=4 -> 2
    """
    return -1 # replace with your code
```

4) The greatest common divisor (GCD) of two values can be computed using Euclid's algorithm. Starting with the values  $m$  and  $n$ , we repeatedly apply the formula:  $n, m = m, n \% m$ , until  $m$  is 0. Then,  $n$  is the GCD of the original values of  $m$  and  $n$ .

Write a function that finds the GCD of two numbers using this algorithm.

```
def gcd(m, n):
    """
    Calculates the GCD of m and n using Euclid's algorithm.

    Process:
    * n = m
    * m = n % m (note this must be simultaneous)
    * continue until m is 0, then the result is in n
    """
    return -1 # replace with your code
```