

Practice Midterm 1 - CS 261

Here are questions similar to what will be on the midterm exam. You won't be able to use any outside material during the midterm exam (no notes, computers, etc.) so see how much you can answer without looking things up.

1. The following code is supposed to print the perfect squares 1, 4, 9, 16, ... up to 100. Circle the mistake in the code and explain why it doesn't do what it is supposed to.

```
n = 1
while n <= 10:
    square = n ** 2
    n + 1
    print(square)
```

Solution: The mistake is in the line `n + 1`. It never changes the value of `n`, so the loop goes on forever.

2. 100 minutes is 1 hour and 40 minutes. How many hours and minutes are in 291 minutes? You don't need to calculate the answer. Just complete the following Python assignment statements with expressions that would calculate the answer.

```
hours =
minutes =
```

Solution:

```
hours = 291 // 60
minutes = 291 % 60
```

3. For each of the following Python expressions, write down the output when the expression is evaluated using a Python interpreter. Write **error** if you think the expression will raise an error.

(a) `15 % 3`

Solution: 0

(b) `5 + 3 * 4 == 100`

Solution: False

(c) `"HSC" + "2024"`

Solution: "HSC2024"

(d) `100+"1"`

Solution: error

(e) $6 / 2$

Solution: 3.0

4. Suppose that we type the following assignments and expressions in a Python shell in the given order.

```
>>> a = 10
>>> b = 20
>>> c = a + b
```

(a) What will Python output if we enter the following (after those first three statements)?

```
>>> a + 5
```

(b) What will Python output if we enter this statement (after the one from part (a))?

```
>>> a + b
```

(c) What will Python output when we enter these two statements next?

```
>>> b = b + 1
>>> c
```

(d) What about if we enter this last?

```
>>> a + b
```

5. Consider the following function. What will it return if you call `mystery(5)`?

```
def mystery(n):  
    a = n  
    a + 5  
    if a > 10:  
        return a  
    elif a == 10:  
        return 100  
    else:  
        return n
```

6. Write a function called `average3` that inputs any three numbers and returns their average.

7. Consider the simple function below.

```
def twice(n):  
    print(2 * n)
```

Why do we get `False` if we enter the following into the shell? Explain why we get the output below.

```
>>> twice(5) == 10  
10  
False
```

8. Describe in words what the following function will do when you call it.

```
def loop_function():  
    for i in range(100):  
        if i % 3 == 0:  
            print(i)
```

9. Need more about types: Identify the type (str, int, float, or bool) for the following expressions.