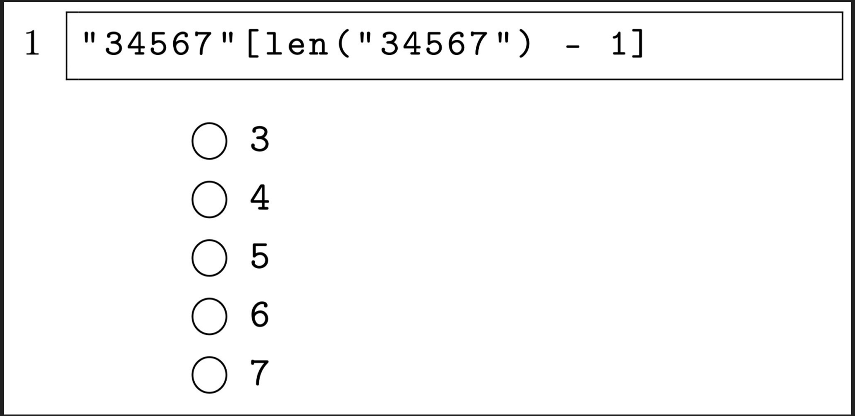
COMP 110

Commonly Missed Quiz Questions

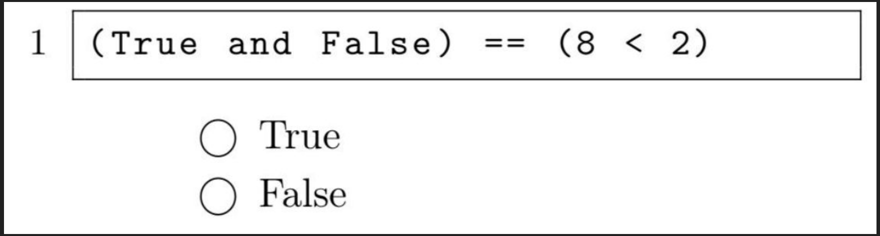
Quiz 00: Commonly Missed Questions



```
"34567"[len("34567") - 1]
```

- ○ 3
- ○ 4
- ○ 5
- ○ 6
- ○ 7

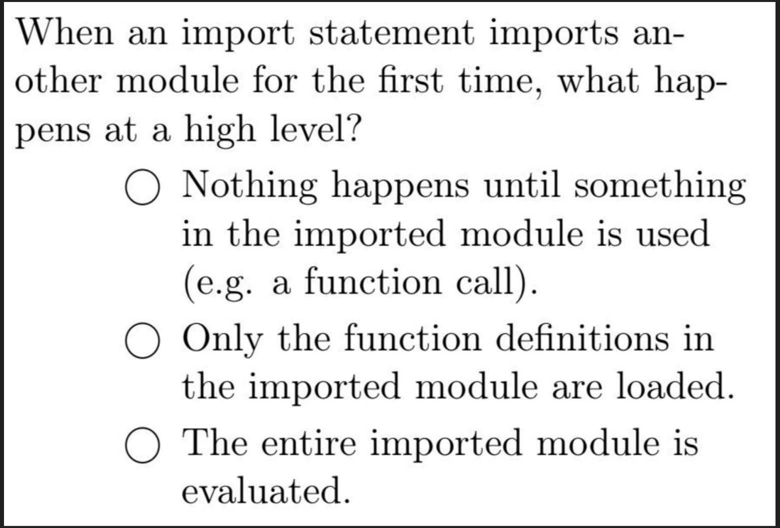
Quiz 01: Commonly Missed Questions



```
(True and False) == (8 < 2)
```

- ◯ True
- ◯ False

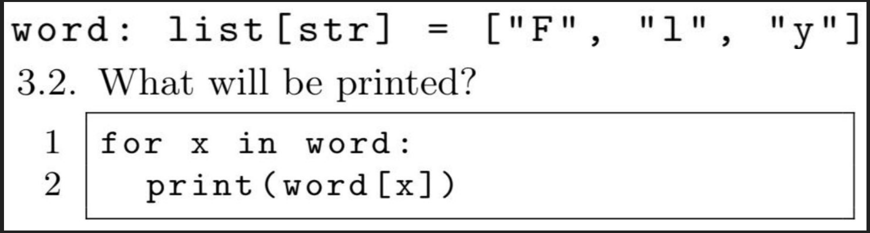
Quiz 02: Commonly Missed Questions



When an import statement imports another module for the first time, what happens at a high level?

- ◯ Nothing happens until something in the imported module is used (e.g. a function call).
- ◯ Only the function definitions in the imported module are loaded.
- ◯ The entire imported module is evaluated.

Quiz 02: Commonly Missed Questions

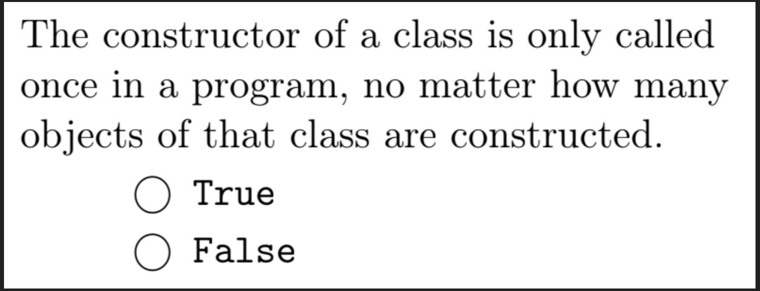


```
word: list[str] = ["F", "l", "y"]
```

3.2. What will be printed?

```
for x in word:
  print(word[x])
```

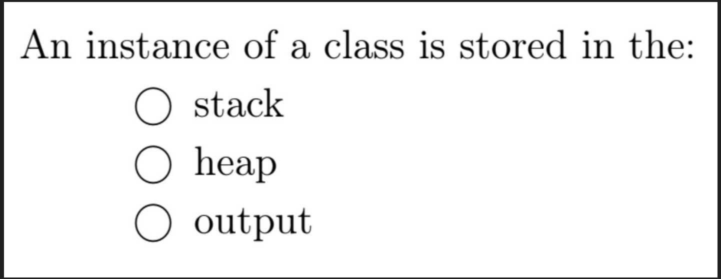
Quiz 03: Commonly Missed Questions



The constructor of a class is only called once in a program, no matter how many objects of that class are constructed.

- ◯ True
- ◯ False

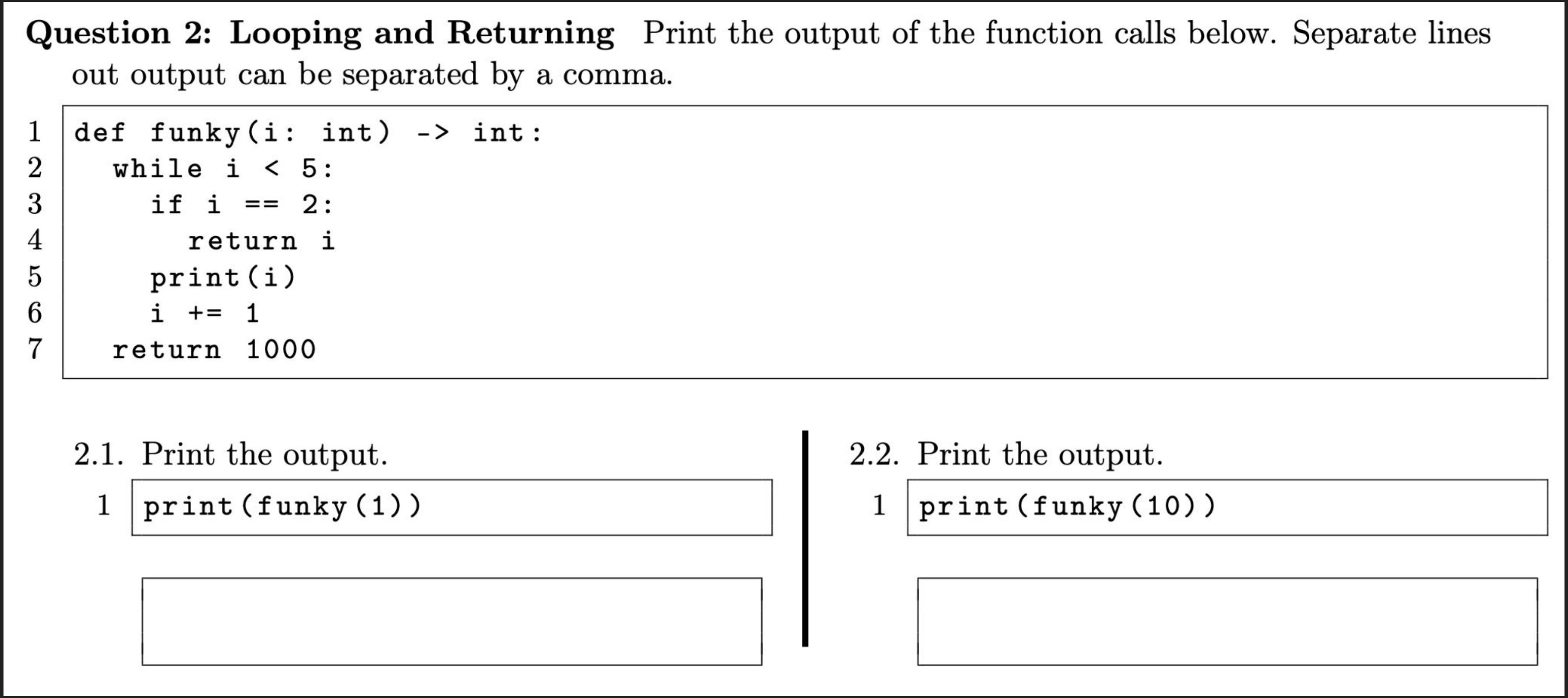
Quiz 03: Commonly Missed Questions



An instance of a class is stored in the:

- ◯ stack
- ◯ heap
- ◯ output

Quiz 03: Commonly Missed Questions



Question 2: Looping and Returning Print the output of the function calls below. Separate lines out output can be separated by a comma.

```
def funky(i: int) -> int:
  while i < 5:
    if i == 2:
      return i
    print(i)
    i += 1
  return 1000
```

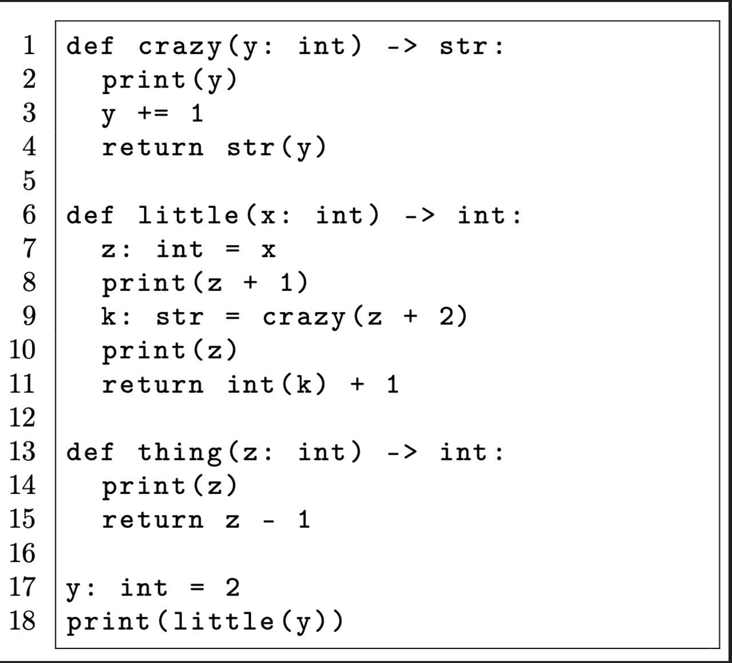
2.1. Print the output.

```
print(funky(1))
```

2.2. Print the output.

```
print(funky(10))
```

Quiz 03: Commonly Missed Questions: Memory Diagram



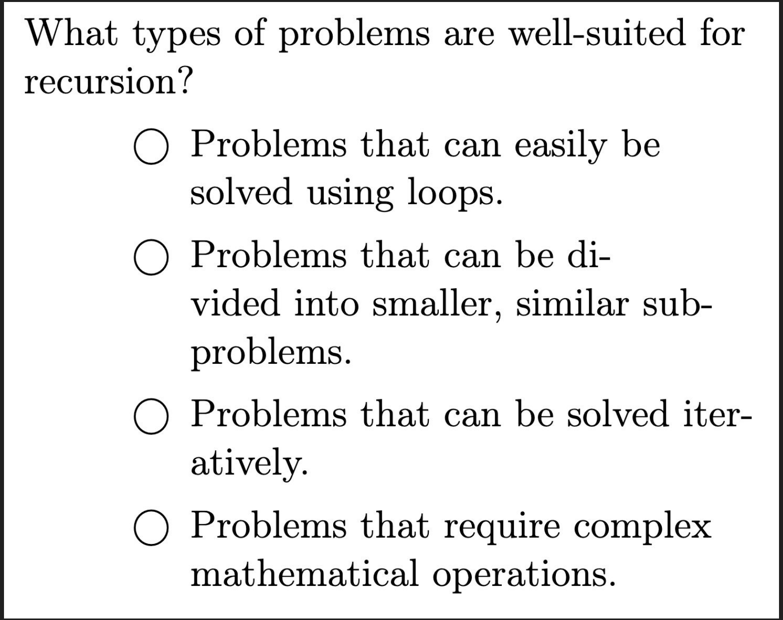
```python
def crazy(y: int) -> str:
  print(y)
  y += 1
  return str(y)

def little(x: int) -> int:
  z: int = x
  print(z + 1)
  k: str = crazy(z + 2)
  print(z)
  return int(k) + 1

def thing(z: int) -> int:
  print(z)
  return z - 1

y: int = 2
print(little(y))
```

Quiz 04: Commonly Missed Questions



What types of problems are well-suited for recursion?

- ◯ Problems that can easily be solved using loops.
- ◯ Problems that can be divided into smaller, similar subproblems.
- ◯ Problems that can be solved iteratively.
- ◯ Problems that require complex mathematical operations.

Memory Diagram

```python
def pie(w: int, k: int) -> int:
    if w < 0 or k < 0:
        raise ValueError("Try again.")

    if w == 0:
        return k
    if k == 0:
        return w

    diff: int = w - k

    if diff > 0:
        return pie(diff - 2, k) - 1
    else:
        return pie(w, k - w) + 3


print(pie(12, 4))
```

Quiz 04: Commonly Missed Questions