

```
y = 3 + 2 + 1
```

```
y = y + 2
```

At the end of this code, what value is assigned to y?

- A. 2
- B. 6
- C. 8
- D. This code will cause an error

Answer: C

After the first line, y is assigned a value of 6. On the second line, y is first evaluated (to 6), before summing and then reassignment

```
y = 3
x = 7
y = 3 + 2 + 1
```

At the end of this code, what will the memory look like?

A. y: 6	B. y: 3 x: 7 y: 6
C. y: 6 x: 7	D. y: 3 x: 7

Answer: C

Both y and x are defined, but each variable, i.e., y, can only have one value at a time

```
x = 6  
print(4x)
```

What would this code print?

- A. 6
- B. 24
- C. 4x
- D. Nothing, this code produces a syntax error

Answer: D

While in mathematical notation we can express multiplication by placing two operators next to each other, in Python (and most programming languages) we need to explicitly express multiplication with the `*` operator. In this case Python tries to parse `4x` as number starting with a 4 digit, e.g., 40, but failing from `x` is not a valid digit.

Credit: Adapted from John Foley under a MIT license

Which of the following does not contain a function call?

A. `type(4.5)`

B. `area(2, 9)`

C. `def add_one(x):
 return x+1`

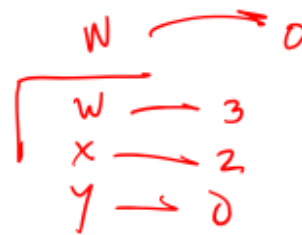
D. `print("hello")`

E. None of the above, all contain function calls

Answer: C

`def ...` is a function definition, not a function call or invocation (call and invocation are synonyms)

```
1 w = 0
  def calculate(w, x, y):
    a = x 2
    b = w + 1 4
    return a + b + 3 9
  calculate(3, 2, 0)
```



What does the call to `calculate` return?

- A. 3
- B. 5
- C. 6
- D. 9

Answer: D

When `calculate` is invoked `w`, `x`, `y` are assigned 3, 2, and 0 respectively. In the function body, `a` and `b` are assigned 2 and 4 respectively. Thus the return value is $2 + 4 + 3$ or 9. The previous value of `w` is not relevant and is not impacted by the call to `calculate`.

```
def add_one(x):  
    return x + 1
```

```
x = 2  
x = x + add_one(x)
```

What are the bugs in the above code?

- A. There are no bugs
- B. The function body is not indented
- C. We are using x as a parameter and variable, but we are not allowed to do that
- D. Both B and C are bugs

Answer: B

Function bodies must be consistently indented. Recall function parameters are only defined within the function body (which the function scope) and can have the same name as other variables.