CSC108H Winter 2024 Worksheet 01 : Arithmetic Operators

1. **Arithmetic Operators**

   Without running the code, complete the table below by filling in the values that the expressions produce and the types of those values.

(a)

Python Expression	Result	Type of Result
`9 / 3`	`3.0`	`float`
`9 // 3`	3	int
`9 % 3`	0	int

(b)

Python Expression	Result	Type of Result
`10 / 4`	2.5	float
`10 // 4`	2	int
`10 // 3`	3	int
`10 % 3`	1	int

2. **Arithmetic Operators**

   For which positive integers `n` does `n % 2` produce `0`? all positive even ints

   For which positive integers `n` does `n % 2` produce `1`? all positive odd ints

3. **Order of Precedence of Arithmetic Operators**

   In the table below, add parentheses to indicate the order that the operations are evaluated.

Python Expression	Python Expression Parenthesized
`-2 + 4 * 7`	`((-2) + (4 * 7))`
`3 + 5 * 2`	(3 + (5 * 2))
`4 + 8 / 2 ** 2 / -2`	(4 + ((8 / (2 ** 2)) / (-2)))

CSC108H Winter 2024 Worksheet 02 : Built-in Functions

1. Built-in Functions

Without running the code, complete the table below by filling in the values that the expressions produce, as well as the types of those values.

Python Expression	Result	Type of Result
`min(4, 6, 2.5)`	`2.5`	`float`
`max(10.1, 13, 16)`	16	int
`abs(-5.2)`	5.2	float
`pow(2, 3)`	8	int

2. Built-in Function: `help`

The built-in function `help` can be used to provide information about other functions. Answer the following questions, using the output of `help(round)` below:

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.  Otherwise
    the return value has the same type as the number.  ndigits may be negative.
```

Question	Answer
What are the types function `round` can return?	int, float
What is the *minimum* number of arguments function `round` can take?	1
What is the *maximum* number of arguments function `round` can take?	2

3. Built-in Function: Using `round`

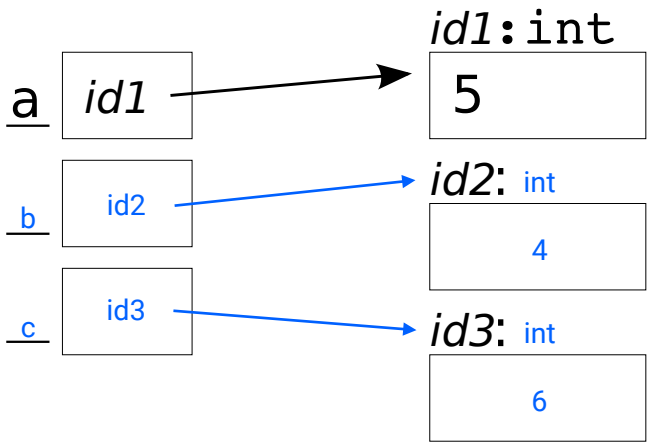
Complete the table below with the results of the following calls to `round`:

Python Expression	Result		Python Expression	Result
`round(1.6)`	2		`round(3.14159, 2)`	3.14
`round(1234.5678, -2)`	1200		`round(2.5)`	2

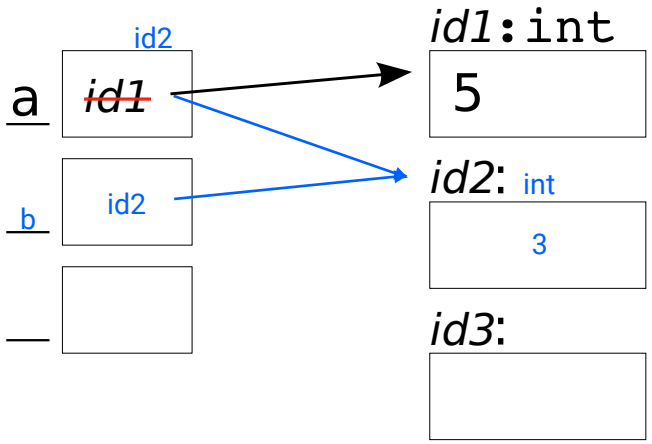
CSC108H Winter 2024 Worksheet 03 : Variable Assignment Memory Model

For each of the following code snippets, fill in the memory model diagram on the right to reflect the state of memory after the code is executed. In each case, we begin with the state where `a` has already been assigned the value `5`. You may not need to use all of the boxes.

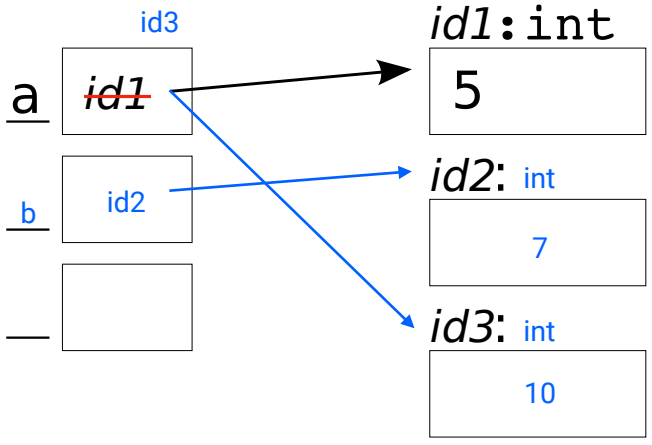
```
a = 5
b = 4
c = 6
```




```
a = 5
a = 3
b = a
```




```
a = 5
b = a + 2
a = 10
```



CSC108H Winter 2024 Worksheet 04 : Variable Assignment Statements

1. **Changing variable values**

   (a) Suppose that you have evaluated some code that sets variable `k` to refer to some value.
   Write an assignment statement that creates a new variable `j` that refers to three times `k`'s value:

   j = 3 * k

   (b) Consider this code:

   ```
   x = 4
   y = 5
   x = 2
   ```

   After the code above is executed, to which value does `x` refer? 2

   After the code above is executed, to which value does `y` refer? 5

   (c) Consider this code:

   ```
   x = 4
   y = x + 2
   x = y + 1
   ```

   After the code above is executed, to which value does `x` refer? 7

   After the code above is executed, to which value does `y` refer? 6

2. **Swapping variable values** *An extra exercise to try at home.*

   Assume that variables `a` and `b` have been assigned `int` values. Write code to swap which values `a` and `b` refer to: after your statements are executed, `a` should refer to the value that `b` used to refer to, and `b` should refer to the value that `a` used to refer to. Hint: use a third variable.

   ```
   c = a
   a = b
   b = c
   ```

   Once you have written the code, trace your code using the memory model to confirm that it correctly swaps the values: