CS100 Study Guide

Topics:

- Value Types (Int/Float/String/Character/Boolean)
- Operations
- Variables
- Comparisons
- While Loops
- If/Elif/Else Statements
- Writing Functions
- Calling Functions
- Function Arguments and Parameters
- Return Values
- Strings
- Arrays
- Dictionaries
- Nested Values (Combinations of arrays and dictionaries)
- Classes
    - You will not be asked to write a class from scratch.

Example Problems:

Expect variable names to be meaningless - don't count on the names telling you what things do. For these examples I've used foo/bar/baz/etc for most of them, to help you practice.

What is printed by the following code?

```
class Foo():
    def __init__(self):
        self.bar = 5

    def baz(self):
        self.bar = self.bar * 2

    def bat(self):
        self.bar = 3
```

```
foo = Foo()
print(foo.bar)
foo.baz()
print(foo.bar)
foo.bat()
print(foo.bar)
```

What is printed?

```
foo = [3, 5, 7, 9]
foo[0] = foo[1] + 1
baz = (3 * 2) - 5
foo.append(baz)
baz = baz - 3
foo.pop(2)
print(foo)
```

What is wrong with this code? Circle the bad code.

```
foo = 100
bar = 10
while bar > 1:
    bar = bar - 2
    foo = foo // bar
    foo = foo * bar
```

Fill in the blanks in this code:

```
# Function to print out an array in reverse order
def reverse_print(array):
    int_i = ________
    while _____________:
        print(_______________)
```

What does this code print?

```
def foo(a):
    a = a - 10
    b = a + 10
    bar(b, a)


def bar(a, b):
    print(a)
    print(b)

int_i = 0
while int_i < 3:
    foo(int_i)
    int_i += 1
```

What does the following code print?

```
class Foo()
    def __init__(self):
        int_i = 0
        while int_i < 3:
            int_j = 0
            while int_j < 3:
                if int_i * int_j % 3 == 0:
                    print(int_i + int_j)

    def add(self, int_i, int_j):
        return int_i + int_j

foo = Foo()
foo.add(3, 3)
```

What is wrong with the following code? Circle the bad code.

```
class Foo():
    def __init__(self, foo):
        foo = foo

    def foo(self, foo):
        self.foo = self.foo + foo

foo = Foo(“foo”)
foo.foo(“foo”)
```

Write a function that will take in a list, reverse it, and return it:

Write a function that will take in an array of strings and remove all strings that contain an ‘a’:

Write a function that will take in an array of strings and return the concatenation of all of them: