1. Basic Python knowledge

For the first question in this assignment, enter the type of the Python expression3.14159 below. Remember that capitalization is important.

float

2. An if statement can have at most how many else parts?

1

3. Consider the following Python code snippet:

def clock\_helper(total\_seconds):

"""

Helper function for a clock

"""

seconds\_in\_minute = total\_seconds % 60

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Enter the value returned by Python after evaluating the expression clock\_helper(90) below.

None

4. In Python, what character always appears at the end of the line before the start of an indented block of code?

:

A colon signals Python that the next statement should be indented.

5. Which of the following expressions returns the last character in the non-empty string my\_string?

my\_string[-1]

my\_string[len(my\_string) - 1]

6. What is the primary difference between a list and a tuple?

List is mutable, tuple is immutable.

7. Consider the following snippet of Python code. What is the value of val2[1] after executing this code?

val1 = [1, 2, 3]

val2 = val1[1:]

val1[2] = 4

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3

8. Which of the following Python expressions is a valid key in a Python dictionary?

None, False, "", 0

Remember that the keys in a Python dictionary must be immutable.

9. Write a function in Python that takes a list as input and repeatedly appends the sum of the last three elements of the list to the end of the list. Your function should loop for 25 times.

def appendsums(lst):

"""

Repeatedly append the sum of the current last three elements

of lst to lst.

"""

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

If your function works correctly, the following code should print 230:

sum\_three = [0, 1, 2]

appendsums(sum\_three)

print sum\_three[10]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Enter the value of sum\_three [20] below.

101902

10. **First, complete the following class definition:**

\_

class BankAccount:

""" Class definition modeling the behavior of a simple bank account """

def \_\_init\_\_(self, initial\_balance):

"""Creates an account with the given balance."""

…

def deposit(self, amount):

"""Deposits the amount into the account."""

…

def withdraw(self, amount):

"""

Withdraws the amount from the account. Each withdrawal resulting

in a negative balance also deducts a penalty fee of 5 dollars

from the balance.

"""

…

def get\_balance(self):

"""Returns the current balance in the account."""

…

def get\_fees(self):

"""Returns the total fees ever deducted from the account."""

…

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

The deposit andn withdraw methods each change the account balance. The withdraw method also deducts a fee of 5 dollars from the balance if the withdrawal (before any fees) results in a negative balance. Since we also have the method get\_fees, you will need to have a variable to keep track of the fees paid.

Here's one possible test of the class. It should print the values 10 and 5, respectively, since the withdrawal incurs a fee of 5 dollars.

my\_account = BankAccount(10)

my\_account.withdraw(15)

my\_account.deposit(20)

print my\_account.get\_balance(), my\_account.get\_fees()

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Copy-and-paste the following much longer test. What two numbers are printed at the end? Enter the two numbers, separated only by spaces.

\_\_

my\_account = BankAccount(10)

my\_account.withdraw(5)

my\_account.deposit(10)

my\_account.withdraw(5)

my\_account.withdraw(15)

my\_account.deposit(20)

my\_account.withdraw(5)

my\_account.deposit(10)

my\_account.deposit(20)

my\_account.withdraw(15)

my\_account.deposit(30)

my\_account.withdraw(10)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.withdraw(50)

my\_account.deposit(30)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.withdraw(5)

my\_account.deposit(20)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.deposit(30)

my\_account.withdraw(25)

my\_account.withdraw(5)

my\_account.deposit(10)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.withdraw(10)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.deposit(30)

my\_account.withdraw(25)

my\_account.withdraw(10)

my\_account.deposit(20)

my\_account.deposit(10)

my\_account.withdraw(5)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.withdraw(5)

my\_account.withdraw(15)

my\_account.deposit(10)

my\_account.withdraw(5)

print my\_account.get\_balance(), my\_account.get\_fees()

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

-60 75