Started on	Monday, 5 August 2024, 1:36 PM
State	Finished
Completed on	Monday, 5 August 2024, 1:42 PM
Time taken	5 mins 31 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

What import module allows a programmer use argparse?

Answer: argparse ✓

The correct answer is: import argparse

Question 2

Correct

Mark 2.00 out of 2.00

True or False: The parameter description cannot be passed into ArgumentParser.

- True
- False ✓

The correct answer is 'False'.

Question 3

Correct

Mark 2.00 out of 2.00

What method allows a programmer to add in arguments to the parser?

Answer: add_argument() ✓

The correct answer is: add_argument

Question 4

Correct

Mark 2.00 out of 2.00

Fill in the blank to finish the code:

```
import argparse

parser = argparse.[ArgumentParse] ✓ ()
parser.add_argument('-r', '--range', help='Allows the programmer to enter a number and create a
range', required=True)

args = parser.parse_args()

if args.range:
    li = []
    for i in range(int(args.range)):
        li.append(i)
print(li)
```

The correct answer is: ArgumentParser

Question 5

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
import argparse, os

parser = argparse.ArgumentParser()
parser.add_argument('-ls', '--ls', help='lists all the files in the current working directory',
required=True)

args = parser.[parse_args] ✓ ()

path = os.getcwd()
files = os.listdir(path)
print(files)
```

The correct answer is: parse_args

Started on	Wednesday, 24 July 2024, 1:47 PM
State	Finished
Completed on	Wednesday, 24 July 2024, 2:07 PM
Time taken	19 mins 15 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

Print two separate items:

1. a floating point number: 24.0
2. The string: "What is funnier than 24?"

Answer: (penalty regime: 0 %)

```
int_number = 24
float_number = float(int_number)
print(float_number)
print("What is funnier than 24?")
```

	Expected	Got	
✓	24.0 What is funnier than 24?	24.0 What is funnier than 24?	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 2

Correct

Mark 2.00 out of 2.00

Using the `type()` function, print the type of `x` the following code block?

Answer: (penalty regime: 0 %)

```
1 x = ''
2 print(type(x))
```

	Expected	Got	
✓	`<class 'str'>`	`<class 'str'>`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 3

Correct

Mark 2.00 out of 2.00

Given `x = 1`, convert `x` to a boolean value and print it.

Answer: (penalty regime: 0 %)

```
x = 1
print(bool(x))
```

	Expected	Got	
✓	`True`	`True`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 4

Correct

Mark 2.00 out of 2.00

True or False: What will the following equate to: `bool(20) and bool(30.0)`?

- (●) True ✓
- ( ) False

The correct answer is 'True'.

Question 5

Correct

Mark 2.00 out of 2.00

Print the addition of 50.0 and 50.

Answer: (penalty regime: 0 %)

```
1 print (50 + 50.0)
2
```

	Expected	Got	
✓	100.0	100.0	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Monday, 5 August 2024, 1:43 PM
State	Finished
Completed on	Monday, 5 August 2024, 1:53 PM
Time taken	10 mins 15 secs
Grade	**12.00** out of 12.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

What will the following code output?

```
strng = 55
assert strng == str(strng), "String is not string."
```

Answer: AssertionError: String is not string. ✓

The correct answer is: AssertionError: String is not string.

Question 2

Correct

Mark 2.00 out of 2.00

Which level of debugging is used by default?

Answer: Warning ✓

The correct answer is: warning

Question 3

Correct

Mark 2.00 out of 2.00

True or False: import logging is used to log in Python.

- (•) True ✓
- ( ) False

The correct answer is 'True'.

Question 4

Correct

Mark 2.00 out of 2.00

```
strng = "Toby, why are you the way you are?"

def func(strng):
    strng = strng.title()
  print(strng)
func(strng)
```

Answer: IndentationError: unindent does not match any outer indentation level ✓

The correct answer is: IndentationError: unindent does not match any outer indentation level

Question 5

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
import logging
logging.basicConfig (level=logging.DEBUG, format=' %(asctime)s - %(levelname)s - %
(message)s')

logging.debug("About to enter while loop")
count = 0
while count <= 10:
    logging.debug(f"{str(count)}")
    count += 1
```

The correct answer is: basicConfig

Question 6

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
try:
    for i in range(num):
        print(i)
except :
    print("Out of range!")
```

The correct answer is: except

Started on	Friday, 26 July 2024, 10:05 AM
State	Finished
Completed on	Friday, 26 July 2024, 10:56 AM
Time taken	51 mins 18 secs
Grade	**20.00** out of 20.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

The ___ loop in Python is used to iterate over the items of any sequence, such as a string, list, dictionary, or any iterable object.

Answer: For ✓

The correct answer is: for

Question 2

Correct

Mark 2.00 out of 2.00

When a continue ✓ statement is executed, the rest of the suite is skipped for that iteration of the loop and control goes to the next iteration.

The correct answer is: continue

Question 3

Correct

Mark 2.00 out of 2.00

True or False: Code inside an `if` suite is only executed if a given condition is `True`.

- (●) True ✓
- ( ) False

The correct answer is 'True'.

Question 4

Correct

Mark 2.00 out of 2.00

What will the following return?

```
x = 0
y = "0"
print(x==y)
```

Answer: False ✓

The correct answer is: False

Question 5

Correct

Mark 2.00 out of 2.00

`not (5 == 5)` resolves to what?

- True
- False ✓

The correct answer is 'False'.

Question 6

Correct

Mark 2.00 out of 2.00

Write a script to iterate through a **range of 0 to 100000**. If the number is **divisible by 2**, **continue**. **Else** add the number to a **count** and **print** out the count.

Answer: (penalty regime: 0 %)

```
count = 0
for i in range(0,100001):
    if i%2==0:
        continue
    else:
        count += i
print(count)
```

	Expected	Got	
✓	2500000000	2500000000	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 7

Correct

Mark 2.00 out of 2.00

Write a python script that finds all the numbers between **3(inclusive) and 5000(exclusive)** that are **evenly divisible by 5** or **evenly divisible by 7** but **not evenly divisible by 35**. **Print the sum.**

Example:
5 10 15 25 30 are divisible by 5 and should be included.

7 14 21 28 are divisible by 7 and should be included

35 70 105 are divisible by 5 and 7 but are also divisible by 35 and should not be included.

Answer: (penalty regime: 0 %)

```
count = 0
for x in range(3, 5000):
    if (x % 35 == 0):
        continue
    elif(x % 7 == 0) or (x % 5 == 0):
        count += x
print(count)
```

	Expected	Got	
✓	3573575	3573575	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00.

Question 8

Correct

Mark 2.00 out of 2.00

Write a for loop to `enumerate` through the list and `start` at 0 below:

```
li = [1,2,3,4,5]
```

Answer: (penalty regime: 0 %)

Reset answer

```
# Write script here
li = [1,2,3,4,5]
for i in enumerate(li):
    print(i)
```

	Expected	Got	
✓	(0, 1) (1, 2) (2, 3) (3, 4) (4, 5)	(0, 1) (1, 2) (2, 3) (3, 4) (4, 5)	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 9

Correct

Mark 2.00 out of 2.00

Write a script to iterate through a range from 1 to 30 and step through every 3rd element. Print it out on one line.

Answer: (penalty regime: 0 %)

```
for i in range(1,31,3):
    print(i, end=" ")
```

	Test	Expected	Got	
✓	`range(1, 30, 3)`	`1 4 7 10 13 16 19 22 25 28`	`1 4 7 10 13 16 19 22 25 28`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 10

Correct

Mark 2.00 out of 2.00

Write a script that has a variable called **count which has the value of 3968** . Using a **range of 1 through 15 both inclusive**, iterate through them. If the element in the range is **not evenly divisible by 3**, **subtract** it from count, and then **print out count**.

Answer: (penalty regime: 0 %)

Reset answer

```
count = 3968
for i in range(1,16):
    if (i % 3 != 0):
        count -= i
print(count)
```

	Expected	Got	
✓	3893	3893	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Monday, 29 July 2024, 11:14 AM
State	Finished
Completed on	Monday, 29 July 2024, 12:20 PM
Time taken	1 hour 5 mins
Grade	**18.00** out of 20.00 (**90**%)

Question 1

Correct

Mark 2.00 out of 2.00

Bill was tasked to split based on username and domain name. Finish the script to accomplish that.

Output:

```
['user', 'domain1.com']
['user', 'domain2.com']
['user', 'domain3.com']
['user', 'domain4.com']
```

Answer: (penalty regime: 0 %)

Reset answer

```
# Write script here.
li = ["user@domain1.com","user@domain2.com","user@domain3.com","user@domain4.com"]
for i in li:
    print(i.split('@'))
```

	Expected	Got	
✓	['user', 'domain1.com'] ['user', 'domain2.com'] ['user', 'domain3.com'] ['user', 'domain4.com']	['user', 'domain1.com'] ['user', 'domain2.com'] ['user', 'domain3.com'] ['user', 'domain4.com']	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 2

Correct

Mark 2.00 out of 2.00

Assign the following two lists to any variable names you like. Add the two lists together and print the combination.

```
[1,2,3,4,5]
["hi","hello","hey"]
```

Answer: (penalty regime: 0 %)

```
num = [1,2,3,4,5]

str = ["hi","hello","hey"]

comb_list = num + str

print(comb_list)
```

	Expected	Got	
✓	`[1, 2, 3, 4, 5, 'hi', 'hello', 'hey']`	`[1, 2, 3, 4, 5, 'hi', 'hello', 'hey']`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 3

Correct

Mark 2.00 out of 2.00

Write a Python script to count the elements from a list of items.

```
MyList = [1,2,"aba", "bab"]
```

Output:

```
1
1
3
3
```

Answer: (penalty regime: 0 %)

Reset answer

```
MyList = [1,2, "aba", "bab", "cab"]
for item in MyList:
    print(len(str(item)))
```

	Expected	Got	
✓	1 1 3 3 3	1 1 3 3 3	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 4

Correct

Mark 2.00 out of 2.00

Write a Python script to sort a dictionary by value.

Answer: (penalty regime: 0 %)

Reset answer

```
di = {5: 1, 4: 3, 2: 2, 6: 9, 3: 4}
sorted_values = sorted(di.values())
print(sorted_values)

```

	Expected	Got	
✓	`[1, 2, 3, 4, 9]`	`[1, 2, 3, 4, 9]`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 5

Correct

Mark 2.00 out of 2.00

Write a script that prints out only unique values from the list.

Answer: (penalty regime: 0 %)

Reset answer

```
li = [1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,5,5,5]
unique_values = set(li)
print(list(unique_values))
```

	Expected	Got	
✓	[1, 2, 3, 4, 5]	[1, 2, 3, 4, 5]	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 6

Correct

Mark 2.00 out of 2.00

The split ✓ method returns a list of the words in a str object.

The correct answer is: split()

Question 7

Incorrect

Mark 0.00 out of 2.00

True or False: `list.sort()` returns a **new** list that is sorted.

- True ✗
- False

The correct answer is 'False'.

Question 8

Correct

Mark 2.00 out of 2.00

Given a dictionary variable named "student_dict":

del student_dict ✓ deletes the entire dictionary. (two words)

The correct answer is: del student_dict

Question 9

Correct

Mark 2.00 out of 2.00

True or False: A tuple is an mutable collection of ordered elements.

- True
- False ✓

The correct answer is 'False'.

Question 10

Correct

Mark 2.00 out of 2.00

Write a script to iterate through a set and print each item of the set on a separate line.

Answer: (penalty regime: 0 %)

Reset answer

```
s = {1,2,3,4,5,6,7}
for item in s:
    print(item)
```

	Expected	Got	
✓	1 2 3 4 5 6 7	1 2 3 4 5 6 7	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Wednesday, 24 July 2024, 2:22 PM
State	Finished
Completed on	Wednesday, 24 July 2024, 2:28 PM
Time taken	5 mins 40 secs
Grade	**12.00** out of 12.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

True or False: What will the following expression evaluate to?

```
bool(5-5)
```

- True
- False ✓

The correct answer is 'False'.

Question 2

Correct

Mark 2.00 out of 2.00

True or False: The addition operator + returns the sum of two numbers.

- True ✓
- False

The correct answer is 'True'.

Question 3

Correct

Mark 2.00 out of 2.00

What will the output be?

```
a = 5
b = 10
c = 15
x = a + b
y = x - c
z = y - x + c
z = bool(z)

print(z)
```

Answer: False ✓

The correct answer is: False

Question 4

Correct

Mark 2.00 out of 2.00

What will the following expression evaluate to?

```
bool(1-1) == bool(0)
```

Answer: True ✓

The correct answer is: True

Question 5

Correct

Mark 2.00 out of 2.00

What does the following code return?

```
ord('%')
```

Answer: 37 ✓

The correct answer is: 37

Question 6

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code to get the following output: `80 121 116 104 111 110`

```
strng = "Python"

for i in strng:
    print(____(i), end=" ")
```

Answer: ord ✓

The correct answer is: ord

Started on	Tuesday, 30 July 2024, 9:52 AM
State	Finished
Completed on	Tuesday, 30 July 2024, 10:44 AM
Time taken	52 mins 16 secs
Grade	**18.00** out of 20.00 (**90**%)

Question 1

Correct

Mark 2.00 out of 2.00

Which python function will read the entire contents of a file in a single string?

Answer: read() ✓

The correct answer is: read

Question 2

Incorrect

Mark 0.00 out of 2.00

What mode is this file opened in?

```
file_obj = open("info.txt")
```

Answer: open() ✗

The correct answer is: r

Question 3

Correct

Mark 2.00 out of 2.00

True or False: If a file exists, write mode will not overwrite the file.

- True
- False ✓

The correct answer is 'False'.

Question 4

Correct

Mark 2.00 out of 2.00

What python function allows a programmer to open a file. This function has designated operations such as read, write, and append.

Answer: open() ✓

The correct answer is: open

Question 5

Correct

Mark 2.00 out of 2.00

readlines() ✓ returns the whole text file in the form of a list.

The correct answer is: readlines()

Question 6

Correct

Mark 2.00 out of 2.00

Using the `with` statement, open users.txt.

Save the data found in the file `users.txt` into a variable named `contents` and then print `contents`

Answer: (penalty regime: 0 %)

```
with open('users.txt', 'r') as file:
    contents = file.read()
print(contents)
```

	Expected	Got	
✓	john.doe jane.smith adam.jones emily.brown michael.wilson alice.smilth bob.johnson sara.williams david.miller lisa.taylor kevin.jackson hannah.anderson peter.davis olivia.martin jack.wilson grace.thompson samuel.roberts natalie.white jacob.hall mia.carter	john.doe jane.smith adam.jones emily.brown michael.wilson alice.smilth bob.johnson sara.williams david.miller lisa.taylor kevin.jackson hannah.anderson peter.davis olivia.martin jack.wilson grace.thompson samuel.roberts natalie.white jacob.hall mia.carter	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 7

Correct

Mark 2.00 out of 2.00

Using users.txt read all of the usernames into a list, where each username is in its own index. Print the list.

Example list (shortened, your list should include all names found in file):

```
['john.doe\n', 'jane.smith\n', 'adam.jones\n', ...]
```

Answer: (penalty regime: 0 %)

```
with open('users.txt', 'r') as file:
    usernames = file.readlines()
print(usernames)
```

	Expected	Got	
✓	`['john.doe\n', 'jane.smith\n', 'adam.jones\n', 'emily.brown\n', 'michael.wilson\n', 'alice.smilth\n', 'bob.johnson\n', 'sara.williams\n', 'david.miller\n', 'lisa.taylor\n', 'kevin.jackson\n', 'hannah.anderson\n', 'peter.davis\n', 'olivia.martin\n', 'jack.wilson\n', 'grace.thompson\n', 'samuel.roberts\n', 'natalie.white\n', 'jacob.hall\n', 'mia.carter']`	`['john.doe\n', 'jane.smith\n', 'adam.jones\n', 'emily.brown\n', 'michael.wilson\n', 'alice.smilth\n', 'bob.johnson\n', 'sara.williams\n', 'david.miller\n', 'lisa.taylor\n', 'kevin.jackson\n', 'hannah.anderson\n', 'peter.davis\n', 'olivia.martin\n', 'jack.wilson\n', 'grace.thompson\n', 'samuel.roberts\n', 'natalie.white\n', 'jacob.hall\n', 'mia.carter']`	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00.

Question 8

Correct

Mark 2.00 out of 2.00

Using the code provided, troubleshoot and find the error(s).

Fix the code to execute properly so that it will print each line of the file on separate lines.

Example output:

This is line 1

This is line 2

`infile` can point to any test file you choose. This question will use the following file for testing purposes: file1.txt

Answer: (penalty regime: 0 %)

```
def file_read(infile):
    with open(infile,'r') as reading:
        li = reading.readlines()
        for line in li:
            print(line)

```

	Test	Expected	Got	
✓	`file_read('file1.txt')`	Why do programmers prefer dark mode? Because light attracts bugs! I would tell you a joke about UDP, but you might not get it. Why did the programmer quit his job? Because he didn't get arrays. I told my wife she should embrace her mistakes. She gave me a hug. Debugging: Removing the needles from the haystack. Why did the developer go broke? Because he used up all his cache. Why don't programmers like nature? It has too many bugs. There are only 10 types of people in the world: those who understand binary, and those who don't. Why was the JavaScript developer sad? Because he didn't know how to 'null' his emotions. Why do Java developers wear glasses? Because they can't C#.	Why do programmers prefer dark mode? Because light attracts bugs! I would tell you a joke about UDP, but you might not get it. Why did the programmer quit his job? Because he didn't get arrays. I told my wife she should embrace her mistakes. She gave me a hug. Debugging: Removing the needles from the haystack. Why did the developer go broke? Because he used up all his cache. Why don't programmers like nature? It has too many bugs. There are only 10 types of people in the world: those who understand binary, and those who don't. Why was the JavaScript developer sad? Because he didn't know how to 'null' his emotions. Why do Java developers wear glasses? Because they can't C#.	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 9

Correct

Mark 2.00 out of 2.00

Using the function provided accomplish the following task:

Read the contents of the file path stored in `infile` and write the contents to the file path stored in `outfile`.

NOTE: The challenge will use file.txt to test your code, however, you can create your own files to accomplish this if you wish.

Answer: (penalty regime: 0 %)

Reset answer

```
def copy_paste(infile, outfile):
    with open(infile, 'r') as file:
        contents = file.read()
    with open(outfile, 'w') as file:
        file.write(contents)

```

	Test	Expected	Got	

✓	`copy_paste('file1.txt','outfile.txt')`
	`def read_outfile(outfile):`
	`  with open(outfile) as outfile:`
	`  print(outfile.read())`
	`read_outfile('outfile.txt')`
		Why do programmers prefer dark mode? Because light attracts bugs!
		I would tell you a joke about UDP, but you might not get it.
		Why did the programmer quit his job? Because he didn't get arrays.
		I told my wife she should embrace her mistakes. She gave me a hug.
		Debugging: Removing the needles from the haystack.
		Why did the developer go broke? Because he used up all his cache.
		Why don't programmers like nature? It has too many bugs.
		There are only 10 types of people in the world: those who understand binary, and those who don't.
		Why was the JavaScript developer sad? Because he didn't know how to 'null' his emotions.
		Why do Java developers wear glasses? Because they can't C#.
			Why do programmers prefer dark mode? Because light attracts bugs!
			I would tell you a joke about UDP, but you might not get it.
			Why did the programmer quit his job? Because he didn't get arrays.
			I told my wife she should embrace her mistakes. She gave me a hug.
			Debugging: Removing the needles from the haystack.
			Why did the developer go broke? Because he used up all his cache.
			Why don't programmers like nature? It has too many bugs.
			There are only 10 types of people in the world: those who understand binary, and those who don't.
			Why was the JavaScript developer sad? Because he didn't know how to 'null' his emotions.
			Why do Java developers wear glasses? Because they can't C#.
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 10

Correct

Mark 2.00 out of 2.00

Use python to finish the function below that will perform the following:

- Read file specified by the path stored in the `inpath` parameter and write all lines to the file specified by the `outpath` parameter.
- Before writing out each line, add the line number starting with 1 follow by colon and space

A test case will utilize users.txt

Answer: (penalty regime: 0 %)

```
def linenums(inpath, outpath):
    with open(inpath, 'r') as in_file, open(outpath, "w") as out_file:
        line_number = 1
        for line in in_file:
            out_file.write(f"{line_number}: {line}")
            line_number += 1


```

	Test	Expected	Got	

✓	`linenums('users.txt','this.txt')`
	`def linenums_test(outpath):`
	`  with open(outpath) as fp:`
	`  print(fp.read())`
	`linenums_test('this.txt')`
		`1: john.doe`
		`2: jane.smith`
		`3: adam.jones`
		`4: emily.brown`
		`5: michael.wilson`
		`6: alice.smilth`
		`7: bob.johnson`
		`8: sara.williams`
		`9: david.miller`
		`10: lisa.taylor`
		`11: kevin.jackson`
		`12: hannah.anderson`
		`13: peter.davis`
		`14: olivia.martin`
		`15: jack.wilson`
		`16: grace.thompson`
		`17: samuel.roberts`
		`18: natalie.white`
		`19: jacob.hall`
		`20: mia.carter`
			`1: john.doe`
			`2: jane.smith`
			`3: adam.jones`
			`4: emily.brown`
			`5: michael.wilson`
			`6: alice.smilth`
			`7: bob.johnson`
			`8: sara.williams`
			`9: david.miller`
			`10: lisa.taylor`
			`11: kevin.jackson`
			`12: hannah.anderson`
			`13: peter.davis`
			`14: olivia.martin`
			`15: jack.wilson`
			`16: grace.thompson`
			`17: samuel.roberts`
			`18: natalie.white`
			`19: jacob.hall`
			`20: mia.carter`
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Wednesday, 31 July 2024, 10:39 AM
State	Finished
Completed on	Wednesday, 31 July 2024, 11:08 AM
Time taken	29 mins 10 secs
Grade	**18.00** out of 20.00 (**90**%)

Question 1

Correct

Mark 2.00 out of 2.00

To define a function, use the keyword def ✓ followed by a name (function name) followed by a parenthesized list of arguments followed by a colon.

The correct answer is: def

Question 2

Correct

Mark 2.00 out of 2.00

True or False: A function can be used (called) before it is defined.

- True
- False ✓

The correct answer is 'False'.

Question 3

Incorrect

Mark 0.00 out of 2.00

True or False: The following function has the correct syntax.

```
def func()
```

- True ✗
- False

The correct answer is 'False'.

Question 4

Correct

Mark 2.00 out of 2.00

function call ✓ calls the function that is defined which will execute what is in the function. (2 words)

The correct answer is: function call

Question 5

Correct

Mark 2.00 out of 2.00

A parameter ✓ is a name used in a function definition. Programmers specify this to be "passed in" as an argument to their function.

The correct answer is: parameter

Question 6

Correct

Mark 2.00 out of 2.00

Write a function to take the list parsed in as argument. Use the provided list to create an odd and even list from the original list. Print them using separate print statements. An example list that will be passed into the function is below. Print the even list first, followed by the odd list.

num = [1,2,3,4,5,6,7,8,9,10]

Answer: (penalty regime: 0 %)

Reset answer

```
def even_odd(num):
    even_numbers = [n for n in num if n % 2 == 0]
    odd_numbers = [n for n in num if n % 2 != 0]
    print( even_numbers)
    print(odd_numbers)
```

	Test	Expected	Got	
✓	even_odd(num = [1,2,3,4,5,6,7,8,9,10])	[2, 4, 6, 8, 10] [1, 3, 5, 7, 9]	[2, 4, 6, 8, 10] [1, 3, 5, 7, 9]	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00.

Question 7

Correct

Mark 2.00 out of 2.00

Given a function and one parameter, `word: str`, print to the screen a single string containing the third, fifth, and seventh character of the string.

Assume the `word` parameter will be at least seven characters long.

Example:

```
bang('CyberSecurity')
```

Should print:

Answer: (penalty regime: 0 %)

Reset answer

```
def bang(word: str) -> None:
    result = word[2] + word[4] + word[6]
    print(result)
```

	Test	Expected	Got	
✓	bang("1234567")	357	357	✓
✓	bang("Donatello")	ntl	ntl	✓
✓	bang("Michelangelo")	cea	cea	✓
✓	bang("Ralphael")	lhe	lhe	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 8

Correct

Mark 2.00 out of 2.00

Given a function with two parameters, `name: str, year: int`, print to the screen a message telling the user how old the `name` will be in the year 2099. **Note:** the `year` provided is the birth year.

Example:

```
getting_old('James', 1999)
```

Should print the following message:

```
James will be 100 in the year 2099.
```

Answer: (penalty regime: 0 %)

Reset answer

```
def getting_old(name: str, year: int) -> None:
    age = 2099 - year
    print(f"{name} will be {age} in the year 2099.")
```

	Test	Expected	Got	
✓	getting_old("Brandon", 1986)	Brandon will be 113 in the year 2099.	Brandon will be 113 in the year 2099.	✓
✓	getting_old("Josh", 1990)	Josh will be 109 in the year 2099.	Josh will be 109 in the year 2099.	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 9

Correct

Mark 2.00 out of 2.00

Given a function that takes two parameters, `(first_name: str, last_name: str`), print out the two parameters **as a single string** with the first letter of each being capitalized. The parameters should be separated by a space.

Example:

```
fullname('james', 'smith')
```

Should print:

```
'James Smith'
```

Answer: (penalty regime: 0 %)

Reset answer

```
def fullname(first_name: str, last_name: str) -> None:
    full_name = f"{first_name.capitalize()} {last_name.capitalize()}"
    print(full_name)
```

	Test	Expected	Got	
✓	`fullname('mike', 'larry')`	Mike Larry	Mike Larry	✓
✓	`fullname('janet', 'willis')`	Janet Willis	Janet Willis	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 10

Correct

Mark 2.00 out of 2.00

Given a function that has two parameters, `number: int, length: int`, print to the screen a string formatted to pad zeros to the left of the `number`. The total number of characters should be equal to the `length`.

Example:

```
zero_pad_number(2, 5)
```

Should print:

```
00002
```

NOTE: Python does not allow leading zeros in integers. What type of object does 00002 need to be?

Answer: (penalty regime: 0 %)

Reset answer

```
def zero_pad_number(number: int, length: int) -> None:
    padded_number = str(number).zfill(length)
    print(padded_number)

```

	Test	Expected	Got	
✓	`zero_pad_number(72, 5)`	`00072`	`00072`	✓
✓	`zero_pad_number(5362, 6)`	`005362`	`005362`	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00.

Started on	Tuesday, 6 August 2024, 2:28 PM
State	Finished
Completed on	Tuesday, 6 August 2024, 4:05 PM
Time taken	1 hour 37 mins
Grade	**16.00** out of 16.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code.

```
class  Person:

    def __init__(self,name, age):
        self.name = name
        self.age = age
```

The correct answer is: class

Question 2

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
class Animal:
  def __init__(self, animal_type, name):
    self.animal_type = animal_type
    self.name = name

animal_one = Animal ("Dog", "Big Mike")

print(animal_one)
```

The correct answer is: Animal

Question 3

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
class Animal:
  def __init__(self, animal_type, name):
    self.animal_type = animal_type
    self.name = name

  def [ __str__ ] (self):
    return f"{self.animal_type}({self.name})"

animal_one = Animal("Dog", "Big Mike")

print(animal_one)
```

The correct answer is: __str__

Question 4

Correct

Mark 2.00 out of 2.00

Fill in the blank to complete the code:

```
class OS:
  def [ __init__ ] (self, OS_type):
    self.OS_type = OS_type

  def __str__(self):
    return f"{self.OS_type}"

OS_one = OS("Linux")

print(OS_one)
```

The correct answer is: __init__

Question 5

Correct

Mark 2.00 out of 2.00

There is a block of code given as a solution, but it is not entirely correct. You need to correct the existing code to get the right solution.

Create a class called `ServiceMember` that stores the branch of `service, name, rank,` and `MOS` of a service member. The class should allow changing the MOS and rank.

For example:

- Creating an instance with `soldier1 = ServiceMember('Army', 'Joe Rogan', 'E6', '11B')` and printing `soldier1` should return: "Service: Army, Name: Joe Rogan, Rank: E6, MOS: 11B".
- Calling `soldier1.changeMOS('11A')` and `soldier1.changeRank('2LT')` and then printing `soldier1` should return: "Service: Army, Name: Joe Rogan, Rank: 2LT, MOS: 11A".

Answer: (penalty regime: 0 %)

```
class ServiceMember:
    def __init__(self, service, name, rank, MOS):
        self.service = service
        self.name = name
        self.rank = rank
        self.MOS = MOS

    def changeMOS(self, new_MOS):
        self.MOS = new_MOS

    def changeRank(self, new_rank):
        self.rank = new_rank

    def __str__(self):
        return f"Service: {self.service}, Name: {self.name}, Rank: {self.rank}, MOS: {self.MOS}"

soldier1 = ServiceMember('Army', 'Joe Rogan', 'E6', '11B')
soldier1.changeMOS('11A')
soldier1.changeRank('2LT')
```

	Test	Expected	Got	

✓	`soldier1 = ServiceMember('Army','Joe Rogan','E6','11B')`
	`print(soldier1)`
	`soldier1.changeMOS('11A')`
	`soldier1.changeRank('2LT')`
	`print(soldier1)`
		`Service: Army, Name: Joe Rogan, Rank: E6, MOS: 11B`
		`Service: Army, Name: Joe Rogan, Rank: 2LT, MOS: 11A`
			`Service: Army, Name: Joe Rogan, Rank: E6, MOS: 11B`
			`Service: Army, Name: Joe Rogan, Rank: 2LT, MOS: 11A`
				✓

✓	`soldier2 = ServiceMember('Army','John Wick','E5','17C')`
	`print(soldier2)`
	`soldier2.changeMOS('17A')`
	`soldier2.changeRank('2LT')`
	`print(soldier2)`
		`Service: Army, Name: John Wick, Rank: E5, MOS: 17C`
		`Service: Army, Name: John Wick, Rank: 2LT, MOS: 17A`
			`Service: Army, Name: John Wick, Rank: E5, MOS: 17C`
			`Service: Army, Name: John Wick, Rank: 2LT, MOS: 17A`
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 6

Correct

Mark 2.00 out of 2.00

Build a simple class called `Dog`.

The class should initialize every instance of `Dog` created to start with the following:

Attribute	Value of attribute:
name	Fido
tricks	empty list

Build a function named `change_name` that accepts a parameter. The parameter provided will be used to overwrite the dog's initialized name.

Build another function named `new_trick` that accepts a parameter. The parameter should be appended to the empty list of tricks that each dog is initialized with.

Build a conversion to string function (hint: Think of the double underscore) that returns a string in the following format:
`"My dog's name is {name}. name can perform the following tricks: {tricks}"`

Answer: (penalty regime: 0 %)

```
class Dog:
  def __init__(self):
    self.name = "Fido"
    self.tricks = []

  def change_name(self, max):
    self.name = max

  def new_trick(self, jump):
    self.tricks.append(jump)

  def __str__(self):
    return f"My dog's name is {self.name}. {self.name} can perform the following tricks: {self.tricks}"
```

	Test	Expected	Got	

✓	`test = Dog()`
	`print(test)`
	`test.change_name('Spot')`
	`test.new_trick('sit')`
	`print(test)`
	`test.new_trick('lay down')`
	`print(test)`
		`My dog's name is Fido. Fido can perform the following tricks: []`
		`My dog's name is Spot. Spot can perform the following tricks: ['sit']`
		`My dog's name is Spot. Spot can perform the following tricks: ['sit', 'lay down']`
			`My dog's name is Fido. Fido can perform the following tricks: []`
			`My dog's name is Spot. Spot can perform the following tricks: ['sit']`
			`My dog's name is Spot. Spot can perform the following tricks: ['sit', 'lay down']`
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 7

Correct

Mark 2.00 out of 2.00

Create a class called `Car`. When initializing a `Car` instance, the user must provide the year, make, model, color, trim, and transmission type, in that order. For example, calling `car1 = Car('2024','Honda','Civic','White','EX','Automatic')` and `print(car1)` should return:

Year: 2024
Make: Honda
Model: Civic
Color: White
Trim: EX
Transmission: Automatic

Answer: (penalty regime: 0 %)

```
class Car:
    def __init__(self, year, make, model, color, trim, transmissiton):
        self.year = year
        self.make = make
        self.model = model
        self.color = color
        self.trim = trim
        self.transmissiton = transmissiton

    def __str__(self):
        return (
            f"Year: {self.year} \n"
            f"Make: {self.make} \n"
            f"Model: {self.model}\n"
            f"Color: {self.color}\n"
            f"Trim: {self.trim}\n"
            f"Transmission: {self.transmissiton}\n")

```

	Test	Expected	Got	

✓	`car1 = Car('2024','Honda','Civic','White','EX','Automatic')`
	`print(car1)`
		`Year: 2024`
		`Make: Honda`
		`Model: Civic`
		`Color: White`
		`Trim: EX`
		`Transmission: Automatic`
			`Year: 2024`
			`Make: Honda`
			`Model: Civic`
			`Color: White`
			`Trim: EX`
			`Transmission: Automatic`
				✓

✓	`car1 = Car('2024', 'Toyota', 'Rav-4','Blue','XLE','Automatic')`
	`print(car1)`
		`Year: 2024`
		`Make: Toyota`
		`Model: Rav-4`
		`Color: Blue`
		`Trim: XLE`
		`Transmission: Automatic`
			`Year: 2024`
			`Make: Toyota`
			`Model: Rav-4`
			`Color: Blue`
			`Trim: XLE`
			`Transmission: Automatic`
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 8

Correct

Mark 2.00 out of 2.00

Create a class named `Chest` that can store a large number of items in to a `list` for an explorer. When initializing a `Chest` instance, it can store a random number of items. The chest has a function called `add` that can store an additional random number of items. It also has a function called `drop` that can remove a random number of items from the chest. The `empty` function removes all the items inside the chest. When you print the chest instance, it will return `Chest: [itemX, itemY, itemZ]`.

For example, by calling `chest1 = Chest('swords', 'bows', 'daggers')` and then `print(chest1)`, it will return `Chest: ['swords', 'bows', 'daggers']`.

Answer: (penalty regime: 0 %)

```
import random

class Chest:
    def __init__(self, *items):
        self.items = list(items)

    def add(self, *items):
        self.items.extend(items)

    def drop(self, *items):
        for item in items:
            self.items.remove(item)
        #num_to_drop = random.randint(1, len(self.items))
        #self.items = self.items[:-num_to_drop]

    def empty(self):
        self.items = []

    def __str__(self):
        return f"Chest: {self.items}"

chest1 = Chest('swords', 'bows', 'daggers')

chest1.add('magic spells', 'maps')

chest1.drop("maps", "bows")

chest1.empty()

```

	Test	Expected	Got	

✓	`chest1 = Chest('swords', 'bows', 'daggers')`
	`print(chest1)`
	`chest1.add('magic spells', 'maps')`
	`print(chest1)`
	`chest1.drop('maps','bows')`
	`print(chest1)`
	`chest1.empty()`
	`print(chest1)`
		`Chest: ['swords', 'bows', 'daggers']`
		`Chest: ['swords', 'bows', 'daggers', 'magic spells', 'maps']`
		`Chest: ['swords', 'daggers', 'magic spells']`
		`Chest: []`
			`Chest: ['swords', 'bows', 'daggers']`
			`Chest: ['swords', 'bows', 'daggers', 'magic spells', 'maps']`
			`Chest: ['swords', 'daggers', 'magic spells']`
			`Chest: []`
				✓

	Test	Expected	Got	

✓	`chest1 = Chest('capes', 'boots', 'gloves')`
	`print(chest1)`
	`chest1.add('shields', 'helmets', 'chest plates', 'gauntlets')`
	`print(chest1)`
	`chest1.drop('boots','helmets')`
	`print(chest1)`
	`chest1.empty()`
	`print(chest1)`
		`Chest: ['capes', 'boots', 'gloves']`
		`Chest: ['capes', 'boots', 'gloves', 'shields', 'helmets', 'chest plates', 'gauntlets']`
		`Chest: ['capes', 'gloves', 'shields', 'chest plates', 'gauntlets']`
		`Chest: []`
			`Chest: ['capes', 'boots', 'gloves']`
			`Chest: ['capes', 'boots', 'gloves', 'shields', 'helmets', 'chest plates', 'gauntlets']`
			`Chest: ['capes', 'gloves', 'shields', 'chest plates', 'gauntlets']`
			`Chest: []`
				✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Wednesday, 24 July 2024, 2:07 PM
State	Finished
Completed on	Wednesday, 24 July 2024, 2:22 PM
Time taken	15 mins 23 secs
Grade	**8.00** out of 8.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

Store the following values in the corresponding variables and print them one by one.

Variable Name	Value
s	"string example"
x	20
y	20.0

Answer: (penalty regime: 0 %)

```
s = "string example"
x = 20
y = 20.0
print(s)
print(x)
print(y)
```

	Expected	Got	
✓	string example 20 20.0	string example 20 20.0	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 2

Correct

Mark 2.00 out of 2.00

Create a variable called `x` and store `1000` in it. Create another variable called `y` and store `582` in it. Create a third variable called `z` with the value `x - y`. Print `z`.

Answer: (penalty regime: 0 %)

```
x = 1000
y = 582
z = x - y
print(z)
```

	Expected	Got	
✓	418	418	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 3

Correct

Mark 2.00 out of 2.00

Create a variable `s` and store the string `"Python"` in it. Multiply the string by 10 and print the result.

Answer: (penalty regime: 0 %)

```
1 s = "Python"
2 print(s * 10)
3
4
```

	Expected	Got
✓	PythonPythonPythonPythonPythonPythonPythonPythonPythonPython	PythonPythonPythonPythonPythonPythonPythonPythonPyt

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00.

Question 4

Correct

Mark 2.00 out of 2.00

Given the following variables, change types and add them as integers and store the value in z, print z.

```
x = "10"

y = 10
```

Answer: (penalty regime: 0 %)

```
1 x = "10"
2 y = 10
3 z = int(x) + y
4 print(z)
```

	Expected	Got	
✓	20	20	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Friday, 2 August 2024, 10:08 AM
State	Finished
Completed on	Friday, 2 August 2024, 10:40 AM
Time taken	32 mins 30 secs
Grade	**15.20** out of 16.00 (**95**%)

Question 1

Correct

Mark 2.00 out of 2.00

True or False: The following import statement allows regex to be used in python: `import re`.

- True ✓
- False

The correct answer is 'True'.

Question 2

Correct

Mark 2.00 out of 2.00

search() ✓ only matches on the first instance.

The correct answer is: search()

Question 3

Correct

Mark 2.00 out of 2.00

What regex character would you use if you wanted to quantify a character 0 or 1 times.

Answer: ? ✓

The correct answer is: ?

Question 4

Correct

Mark 2.00 out of 2.00

True or False: Groups() returns all the find elements as a list.

- True
- False ✓

The correct answer is 'False'.

Question 5

Correct

Mark 2.00 out of 2.00

Fill in the blank to finish the code:

```
phone = re.compile(__'\d\d\d-\d\d\d-\d\d\d\d')
```

Answer: r ✓

The correct answer is: r

Question 6

Correct

Mark 2.00 out of 2.00

Write a script to search for the email in the given string.

Answer: (penalty regime: 0 %)

Reset answer

```python
import re

def find_email(string):
    pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
    email = re.search(pattern, string)
    if email:
        return email.group()
    else:
        return "No email found"

strng = "Bill has an email address bill@email.com"
print(find_email(strng))
```

	Expected	Got	
✓	bill@email.com	bill@email.com	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 7

Correct

Mark 1.20 out of 2.00

From the given string, use regex to find and print all emails with a yahoo domain and a three letter first name.

Answer: (penalty regime: 10, 20, ... %)

Reset answer

```
import re

strng = '''Bill has an email address Bill@email.com
           Suzanne has an email address Suzanne@other.org
           Joe has an email address Joe@yahoo.com
           Jill has an email address Jill@yahoo.com
           Jim has an email address Jim@yahoo.com
           Dre has an email address Dre@google.com
           Han has an email address Han@yahoo.com
           Jen has an email address Jennifer@yahoo.com
           '''

pattern = (r'\b([a-zA-Z]{3})\b.*?\1@yahoo\.com')

matches = re.findall(pattern, strng)

print([f"{match}@yahoo.com" for match in matches])
```

	Expected	Got	
✓	['Joe@yahoo.com', 'Jim@yahoo.com', 'Han@yahoo.com']	['Joe@yahoo.com', 'Jim@yahoo.com', 'Han@yahoo.com']	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Python3)**

Correct

Marks for this submission: 2.00/2.00. Accounting for previous tries, this gives **1.20/2.00**.

Question 8

Correct

Mark 2.00 out of 2.00

From the given string, find all the phone numbers regardless of format.

Answer: (penalty regime: 0 %)

Reset answer

```
import re

strng = '''(212)555-1234
           (123)45-2345
           415-555-6789
           123-123-123
           (305)555-4321
           818-555-9876
           (312)555-8765
           '''

pattern = re.compile(r'\(?\d{3}\)?[-\s]?\d{3}[-\s]?\d{4}')

matches = pattern.findall(strng)

print(matches)
```

	Expected	Got	
✓	`['(212)555-1234', '415-555-6789', '(305)555-4321', '818-555-9876', '(312)555-8765']`	`['(212)555-1234', '415-555-6789', '(305)555-4321', '818-555-9876', '(312)555-8765']`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Started on	Thursday, 25 July 2024, 1:03 PM
State	Finished
Completed on	Thursday, 25 July 2024, 1:49 PM
Time taken	46 mins 25 secs
Grade	**20.00** out of 20.00 (**100**%)

Question 1

Correct

Mark 2.00 out of 2.00

Write a script to convert the following string to a list of words.

```
string = "Turn this into a list."
```

Answer: (penalty regime: 0 %)

```
1 list = ['Turn', 'this', 'into', 'a', 'list.']
2 print(list)
```

	Expected	Got	
✓	`['Turn', 'this', 'into', 'a', 'list.']`	`['Turn', 'this', 'into', 'a', 'list.']`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 2

Correct

Mark 2.00 out of 2.00

Write a script to convert an int into hexadecimal. Format the string to look like: `The hexidecimal of {5} is {0x5}.`

Answer: (penalty regime: 0 %)

```
num = 5
hex(num)

print(f"The hexidecimal of {5} is {hex(num)}")
```

	Expected	Got	
✓	`The hexidecimal of 5 is 0x5`	`The hexidecimal of 5 is 0x5`	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 3

Correct

Mark 2.00 out of 2.00

Change the 3rd index to t of a list and then convert it into a string: `["F","e","e","d"]`

Answer: (penalty regime: 0 %)

```
my_str = ["F","e","e","d"]

my_str[3] = "t"

ver = ''.join(my_str)

print(ver)
```

	Expected	Got	
✓	Feet	Feet	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 4

Correct

Mark 2.00 out of 2.00

Replace the letter a, with b. Count how many b's are in the string.

`string: "abbbabbbaaaabaabaababcabababaaabaababcabababbabbbcaabababacbababababababababababa"`

Answer: (penalty regime: 0 %)

Reset answer

```
1 string = "abbbabbbaaaabaabaababcabababaaabaababcabababbabbbcaabababacbababababababababababa"
2
3 print(string.replace("a", "b").count("b"))
```

	Expected	Got	
✓	77	77	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.

Question 5

Correct

Mark 2.00 out of 2.00

What built-in python function will remove any leading and trailing white-space in a string?

Answer: strip() ✓

The correct answer is: strip()

Question 6

Correct

Mark 2.00 out of 2.00

True or False: Strings in Python are mutable.

- True
- False ✓

The correct answer is 'False'.

Question 7

Correct

Mark 2.00 out of 2.00

The str.format() ✓ **method uses curly braces in a template string to mark where the output should be replaced.**

The correct answer is: format

Question 8

Correct

Mark 2.00 out of 2.00

True or False: Literal string values may be enclosed in single or double quotes and can span multiple lines in several ways.

- True ✓
- False

The correct answer is 'True'.

Question 9

Correct

Mark 2.00 out of 2.00

Strings in Python are represented using a class named str ✓ .

The correct answer is: str

Question 10

Correct

Mark 2.00 out of 2.00

Based on the variable below, write a script to round the number 3 decimal places.

```
x = 10.23654785
```

Answer: (penalty regime: 0 %)

Reset answer

```
# Write script here
x = 10.23654785
x = round(10.23654785, 3)
print(x)
```

	Test	Expected	Got	
✓	f'{x:.3f}'	10.237	10.237	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 2.00/2.00.