Back to 'Lab 5 - Lists & Strings'

| | |
|---|---|
| **Started on** | Saturday, 20 October 2018, 5:48 PM |
| **State** | Finished |
| **Completed on** | Saturday, 20 October 2018, 6:05 PM |
| **Time taken** | 16 mins 44 secs |
| **Grade** | **9.00** out of 9.00 (**100**%) |

**Question 1**

Correct

Not graded

Enter your partner's UD email address. If you did not work with a partner enter the word "none".

Answer: bobbymcc@udel.edu

**Question 2**

Correct

Mark 0.50 out of 0.50

Write a statement that assigns an empty list to the variable **empty**.

Answer:

```
1  empty = []
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `print(empty==[])` | `True` | `True` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 3**

Correct

Mark 0.50 out of 0.50

Write a statement which assigns to the variable **odds** all the odd integer values between 0 and 10 (in order).

Answer:

```
1  odds = [1,3,5,7,9]
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `print(odds==[1,3,5,7,9])` | `True` | `True` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 4**

Correct

Mark 0.50 out of 0.50

Write a statement which assigns the list of three-letter abbreviations of the months of the year to the variable **months**. That would be:

- Jan
- Feb
- Mar
- Apr
- May
- Jun
- Jul
- Aug
- Sep
- Oct
- Nov
- Dec

Capitalize correctly and have no punctuation.

Answer:

```
1  months = ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"]
```

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 5**

Correct

Mark 0.50 out of 0.50

Create a list called **threes** which has the values (in this order):

- three
- 3
- 3.0

Answer:

```
threes = ["three",3,3.0]
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `print(threes==['three',3,3.0])` | `True` | `True` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 6**

Correct

Mark 0.50 out of 0.50

Create a list called **coins** with the values 0.01, 0.05, 0.1, 0.25, 0.5, and 1.0.

Answer:

```
coins = [0.01, 0.05, 0.1, 0.25, 0.5, 1.0]
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `print(coins==[.01,.05,.1,.25,.5,1.])` | `True` | `True` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 7**

Correct

Mark 0.50 out of 0.50

Given a list (called **lst**) of six integers, write an expression for the value of the first entry.

Answer:

```
1  lst[0]
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `lst=[0,4,1,3,7,2]` | `0` | `0` | ✓ |
| ✓ | `lst=[9, 1,2,3,4,-9]` | `9` | `9` | ✓ |
| ✓ | `lst=[-9, 1,2,3,4,-9]` | `-9` | `-9` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 8**

Correct

Mark 0.50 out of 0.50

Given a list (called **lst**) of six integers, change the third value to 17.

Answer:

```
1  lst[2] = 17
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `lst=[0,4,1,3,7,2]` | `[0, 4, 17, 3, 7, 2]` | `[0, 4, 17, 3, 7, 2]` | ✓ |
| ✓ | `lst=[0, 1,2,3,4,-9]` | `[0, 1, 17, 3, 4, -9]` | `[0, 1, 17, 3, 4, -9]` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 9**

Correct

Mark 0.50 out of 0.50

Write an expression for the value of the last entry in a list (called **lst**) of six integers,

Answer:

```
1  lst[-1]
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `lst=[0,4,1,3,7,2]` | 2 | 2 | ✓ |
| ✓ | `lst=[9, 1,2,3,4,-9]` | -9 | -9 | ✓ |
| ✓ | `lst=[-9, 1,2,3,4,9]` | 9 | 9 | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 10**

Correct

Mark 0.50 out of 0.50

Given a list (called **lst**) of six integers, add three to the last value.

Answer:

```
1  lst[-1] = lst[-1] + 3
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `lst=[6,4,1,3,7,2]` | `[6, 4, 1, 3, 7, 5]` | `[6, 4, 1, 3, 7, 5]` | ✓ |
| ✓ | `lst=[1,1,2,3,4,-9]` | `[1, 1, 2, 3, 4, -6]` | `[1, 1, 2, 3, 4, -6]` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 11**

Correct

Mark 0.50 out of 0.50

You've been provided a list called **myLst,** of six integers. Set the fourth value to 7.

Answer:

```
1  myLst[3] = 7
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `myLst=[0,4,1,3,7,2]` | `[0, 4, 1, 7, 7, 2]` | `[0, 4, 1, 7, 7, 2]` | ✓ |
| ✓ | `myLst=[0, 1,2,3,4,-9]` | `[0, 1, 2, 7, 4, -9]` | `[0, 1, 2, 7, 4, -9]` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 12**

Correct

Mark 0.50 out of 0.50

A list called **theLst** has six string values. Append "hi" to the first value in the list.

Answer:

```
1  theLst[0] = theLst[0] + "hi"
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `theLst= ['a','b','c','d','e','f']` | `['ahi', 'b', 'c', 'd', 'e', 'f']` | `['ahi', 'b', 'c', 'd', 'e', 'f']` | ✓ |
| ✓ | `theLst= ['q','b','c','d','e','f']` | `['qhi', 'b', 'c', 'd', 'e', 'f']` | `['qhi', 'b', 'c', 'd', 'e', 'f']` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 13**

Correct

Mark 0.50 out of 0.50

Find and print the length of the list **listTest.**

Answer:

```
1  print(len(listTest))
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `listTest=['c','i','s','c','1','0','6']` | 7 | 7 | ✓ |
| ✓ | `listTest=['c','i','s','c','1']` | 5 | 5 | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 14**

Correct

Mark 0.50 out of 0.50

We can test if an item exists in a list or not, using the keyword `in`

It will give you a boolean response if a certain element is present. Find whether 'p' or 'P' is present in **listTest**, and **print** the results. Test for both 'p' and 'P' in order and **print** both results.

**For example:**

| Test | Result |
|---|---|
| `listTest=['c','i','s','c','1','0','6','P','y','t','h','o','n']` | `False`<br>`True` |

Answer:

```
1  print("p" in listTest)
2  print("P" in listTest)
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `listTest=['c','i','s','c','1','0','6','P','y','t','h','o','n']` | `False`<br>`True` | `False`<br>`True` | ✓ |
| ✓ | `listTest=['c','i','s','c','1','0','6','p','y','t','h','o','n']` | `True`<br>`False` | `True`<br>`False` | ✓ |
| ✓ | `listTest=['c','i','s','c','1','0','6','y','t','h','o','n']` | `False`<br>`False` | `False`<br>`False` | ✓ |
| ✓ | `listTest=['c','i','s','c','p','1','0','6','P','y','t','h','o','n']` | `True`<br>`True` | `True`<br>`True` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 15**

Correct

Mark 0.50 out of 0.50

We can delete one or more items from a list using the keyword `del`. It can even delete the list entirely.

From the list **listTest**, delete the 4th element.

Answer:

```
1  del(listTest[3])
```

| | Test | Expected | Got |
|---|---|---|---|
| ✓ | `print(listTest)` | `['c', 'i', 's', '1', '0', '6', 'P', 'y', 't', 'h', 'o', 'n']` | `['c', 'i', 's', '` |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 16**

Correct

Mark 0.50 out of 0.50

Using the `del` keyword and/or the slicing operator `:`, delete the 3rd to 6th elements from the list **listTest**.
**Note**: this does **not** mean the elements at index 3, 4, 5, and 6!

Answer:

```
1  del(listTest[2:6])
```

| | Expected | Got | |
|---|---|---|---|
| ✓ | `['c', 'i', '6', 'P', 'y', 't', 'h', 'o', 'n']` | `['c', 'i', '6', 'P', 'y', 't', 'h', 'o', 'n']` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 17**

Correct

Mark 0.50 out of 0.50

We can access a range of items in a list by using the slicing (colon) operator.

There is a predefined list **listTest.** Use the slicing operator to **print** the third to fifth elements from **listTest** as a list.
**Note:** this does **not** mean the values at indices 3, 4, and 5!

Answer:

```
1 print(listTest[2:5])
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | `listTest=['c','i','s','c','1','0','6']` | `['s', 'c', '1']` | `['s', 'c', '1']` | ✓ |
| ✓ | `listTest=[1,6,7,9,15,-1,'6']` | `[7, 9, 15]` | `[7, 9, 15]` | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 18**

Correct

Mark 0.50 out of 0.50

We can access a range of items in a list by using the slicing operator (colon).

There is a predefined list **listTest.** Use the slicing operator to print the elements (as a list) from beginning through the sixth from **listTest**

Answer:

```
1 print(listTest[:6])
```

| | Test | Expected | Got |
|---|---|---|---|
| ✓ | `listTest= ['c','i','s','c','1','0','6','P','y','t','h','o','n']` | `['c', 'i', 's', 'c', '1', '0']` | `['c', 'i'` |
| ✓ | `listTest= [1,'i',6,'c',-1,'0','6','P','y','t','h','o','n']` | `[1, 'i', 6, 'c', -1, '0']` | `[1, 'i',` |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.

**Question 19**

Correct

Mark 0.50 out of 0.50

We can access a range of items in a list by using the slicing operator (colon).

Given a predefined list **listTest,** use slicing to print the elements from the fourth through the end from **listTest**

Answer:

```
1  print(listTest[3:])
```

| | Test | Expected |
|---|---|---|
| ✓ | `listTest= ['c','i','s','c','1','0','6','P','y','t','h','o','n']` | `['c', '1', '0', '6', 'P', 'y', 't', 'h',` |
| ✓ | `listTest= ['c','i','s','c','1','0','6','P','y','t','h']` | `['c', '1', '0', '6', 'P', 'y', 't', 'h']` |

Passed all tests! ✓

**Correct**

Marks for this submission: 0.50/0.50.