SAMPLE QUESTIONS

1) (15 points) Write a for loop that counts from 5 to -5 in increments of 2. Each number and its square (not square root) should be displayed in the shell window.

2) (15 points) Write a code snippet that gets input from the user. The only valid inputs are shown on the number line below. If the user enters an invalid number, the program insists that the user try again. When the user complies, the number they entered is displayed to the shell window.

3) TRUE/FALSE

a) In the code below, the value of aList[0] will be 5 after the code is executed. T / F

```python
aList = [5, 4, 3, 2, 1]
change(aList)
def change(a):
a[0] += 2
```

b) In Python, functions can be passed lists as arguments and can return lists. T / F

c) The code below will correctly count how many words begin with the letter “G”. T / F

```python
names = ["George", "Elias", "Zayna", "Edgar", "Deanna", "Elizabeth", "Ginny"]
c = 0
for n in names:
    if n[0] == "G":
        c = c + 1
```

d) In the code sample below, the variable i is used as a loop variable. T / F

```python
r = range(0, 10)
for i in r:
    print(i)
```
e) In Python, some data types, like the string, have methods that operate on them. An example of this is shown below:

```
fruit = "banana"
print ( fruit.upper() )
```

4) (20 points) Show how the selection sort will sort this list in descending order.

<table>
<thead>
<tr>
<th>Original List</th>
<th>After pass 1</th>
<th>After pass 2</th>
<th>After pass 3</th>
<th>After pass 4</th>
<th>After pass 5</th>
<th>After pass 6</th>
<th>After pass 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-9.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5) Write a function that removes all the odd numbers from a list that you pass to it. The function does not return anything. Here is how to call it:

```
aList = [ 3, 4, 88, 0, 9, 12, 22 ]
removeOdd( aList )
print (aList)  # should display [ 4, 88, 0, 12, 22]
```

4) (20 points) For each expression, supply the correct data type and value. Use the variable assignments that are given. If the expression produces an error, write ERROR in both columns.

```
x = 3
y = -2.5
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>aList[x]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y[x]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aList</td>
<td>x + y</td>
<td>&quot;I want a &quot; + aList[ int( x + y ) ] + &quot; coffee&quot;</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
</tbody>
</table>