1 Fall 2017 CS 1110/1111 Exam 1

Bubble in your computing ID in the footer of this page. We use an optical scanner to read it, so fill in the bubbles darkly. If you have a shorter ID, leave some rows blank.

In case we have trouble with the scanner, please also legibly print

Your name: ____________________________________________

Your computing ID: _______________________

Please observe the following directions throughout the exam:

- Write legibly; we deduct points if we are unsure what you wrote.
- Indentation and punctuation do matter.
- Write on the lines, where possible.
- If you need to insert a line between two you’ve written, make it clear that that is what you are doing.
- We grade one page at a time. Do not spill answers onto another page.
- Don’t add features we didn’t request: only print if we ask you to print, etc.

The exam is being given in multiple locations simultaneously, so we cannot fairly answer student questions during the exam. If you find a question ambiguous or unclear, write that down on your exam and we’ll give it due consideration during grading.

1.1 Pledge [3 pts]

On my honor as a student, I have neither given nor received help or assistance on this exam.

Signed: ____________________________________________

1.2 Bubble [3 pts]

Enter your computing ID, one symbol per row, top-to-bottom; if you have a short ID (like up3f) leave a row blank on the appropriate side of the digit row.
2 Question [14 pts]

Fill in the following table. If the code does not print and/or does not return, write “None” in the appropriate spot. If the expression results in an error, write “Error” in either column and leave the other column empty. Write values in a way that would have the correct type in Python (e.g., 1 is an int, "1" a str, etc.).
Recall that math.sqrt is the square-root function. Assume that every time the user is required to type something, they type 2.7

<table>
<thead>
<tr>
<th>Code</th>
<th>Prints</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>print(&quot;F&quot; + &quot;un&quot; * 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print(&quot;Fun&quot; + 1 * 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>math.sqrt(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int(&quot;3.14&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float(input(&quot;x: &quot;))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>input(&quot;Fun&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int(float(input(&quot;x: &quot;)))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 Question [10 pts]

The Heisenberg uncertainty principle relates the standard deviation of position and momentum:

\[
\sigma_x \sigma_p \geq \frac{\hbar}{2}
\]

must always be true,

where \(\sigma_x\) is the standard deviation of position, \(\sigma_p\) the standard deviation of momentum, and \(\frac{\hbar}{2}\) is a constant approximately equal to \(10^{-34}\).

Write a function physically_impossible that takes two arguments, sigma_x and sigma_p and returns a Boolean value that is True if the two are physically impossible according to Heisenberg.
4 Question [12 pts]

What does the following code print? If it has an error, write “Error”

def is_int(n):
    if type(n) == type(1):
        return True
    else:
        print("False")

x = 7.0
y = 4
z = 12345
print(is_int(y * x))
print(is_int(y * z))

5 Question [12 pts]

Write some Python code that will set x to be 3 if and only if z is even and $z \leq y \leq 2z$; otherwise, print "something went wrong" and set x to be 0
6 Question [5 pts]

What does the following code print? If it has an error, write “Error”

```python
x = 7
def add_four(x):
    y = x + 4
add_four(4)
print(y)
```

Prints:

7 Question [5 pts]

What does the following code print? If it has an error, write “Error”

```python
x = 'global'
def f(x):
    if len(x) < 8:
        x = x + x
    else:
        x = 'long'
f(x)
f(x)
print(x)
```

Prints:

8 Question [16 pts]

Fill in the following table. The first row is done for you.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 + 2.0</td>
<td>3.0</td>
<td>float</td>
</tr>
<tr>
<td>7 % 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>False and True</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 / 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 * &quot;2&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int(&quot;8&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 &lt; 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 // 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;9&quot; * 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Write a function named `perimeter` such that execution of these Python statements:

```python
print(perimeter("triangle", 2))
print(perimeter("square", 5))
print(perimeter("pentagon", 10))
```

produces the output:

The triangle's perimeter is 6.
The square's perimeter is 20.
The pentagon's perimeter is 50.

You only need to handle those three shapes (triangle, square, and pentagon), but should handle any edge length (not just 2, 5, and 10). Your solution should contain the `if` keyword exactly once.