1 Exercises

Exercise 1. If $s$ is a string, $s.upper()$ returns a copy of $s$ converted to uppercase.

a. What does the following code fragment write?

```python
s = 'Hello World'
s.upper()
stdio.writeln(s[6:11])
```

b. What does the following code fragment write?

```python
s = 'Hello World'
s = s.upper()
stdio.writeln(s[6:11])
```

Exercise 2. Suppose we have a user-defined data type called Circle that represents a circle of radius $r$ centered at $(h, k)$ and supports the following API:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle(h, k, r)</td>
<td>constructs a circle $c$ of radius $r$ centered at $(h, k)$; when no arguments are given, $c$ is a unit circle centered at the origin</td>
</tr>
<tr>
<td>c.area()</td>
<td>returns the area of $c$</td>
</tr>
<tr>
<td>c.contains(x, y)</td>
<td>returns True if $c$ contains $\uparrow (x, y)$ and False otherwise</td>
</tr>
<tr>
<td>c &lt; d</td>
<td>returns True if $c$ is area-wise smaller than $d$, and False otherwise</td>
</tr>
<tr>
<td>c == d</td>
<td>returns True if $c$ and $d$ represent the same circle, and False otherwise</td>
</tr>
<tr>
<td>str(c)</td>
<td>returns a string representation of $c$, as $(h, k, r)$</td>
</tr>
</tbody>
</table>

\[ A \text{ point } (x, y) \text{ is contained in a circle of radius } r \text{ centered at } (h, k) \text{ if } (x - h)^2 + (y - k)^2 \leq r^2 \]

a. Is the Circle data type immutable?

b. How do you create a Circle object $c_1$ representing a circle centered at $(1, 1)$ and having radius 2?

c. How do you create a Circle object $c_2$ representing a unit circle centered at the origin?

d. How do you obtain the area of $c_1$?

e. How do you check if the point $(1.2, 2.2)$ is contained in $c_1$?

f. How do you compare the areas of two circles represented by Circle objects $c$ and $a$ without invoking the area() method explicitly? What does the code translate to internally?

g. How do you check if two Circle objects $c$ and $a$ represent the same circle? What does the code translate to internally?

h. How do you obtain the string representation of $c_1$? What does the code translate to internally?

i. Provide code that creates a list $a$ of 100 Circle objects, each representing a circle centered at the origin and having a random radius from the interval $[0, 1)$.

j. Provide an expression that uses map and reduce to calculate the sum of the areas of the circles stored in the list $a$ from the previous part.

Exercise 3. Write a program called filter.py that accepts three floats $h$, $k$, and $r$ as command-line arguments, creates a Circle object $c$ representing a circle centered at $(h, k)$ and having radius $r$, reads in pairs $(x, y)$ of floats from standard input representing points on a 2D plane, and writes the fraction of points that fall inside the circle $c$. For example

```
$ python3 filter.py 0 0 3
1 2
3 4
1 5
```
2 Solutions to Exercises

Solution 1.

a. World

b. WORLD

Solution 2.

a. Yes

b. \( c_1 = \text{Circle}(1, 1, 2) \)

c. \( c_2 = \text{Circle}() \) OR \( c_2 = \text{Circle}(0, 0, 1) \)

d. \( c_1.\text{area()} \)

e. \( c_1.\text{contains}(1.2, 2.2) \)

f. \( c < d \) which translates to \( \text{c.__lt__(d)} \) internally

g. \( c == d \) which translates to \( \text{c.__eq__(d)} \) internally

h. \( \text{str(c1)} \) which translates to \( \text{c1.__str__()} \) internally

i.

```python
circles = []
for i in range(100):
c = Circle(r = stdrandom.uniform(0, 1))
circles.append(c)
```

j. \( \text{reduce(lambda x, y: x + y, map(lambda x: x.\text{area()}, circles))} \)

Solution 3.

```python
# filter.py
import stdio
import sys
from circle import Circle

def main():
    h = float(sys.argv[1])
k = float(sys.argv[2])
r = float(sys.argv[3])
c = Circle(h, k, r)
total, inside = 0, 0
while not stdio.isEmpty():
x = stdio.readFloat()
y = stdio.readFloat()
total += 1
inside += 1 if c.contains(x, y) else 0
stdio.writeln(1.0 * inside / total)

if __name__ == '__main__':
    main()
```