

Spring, 2017

Name: \_\_\_\_\_

(Please *don't* write your id number!)

# Exam 1: Python Programming

## Directions

For this exam you are permitted one page of notes. It is a good idea to condense your notes into a small amount of ready reference material.

If you need more space, use the back of a page. Note when you do that on the front.

Before you begin, please take a moment to look over the entire test so that you can budget your time.

Clarity is important; if your answers are sloppy and hard to read, you may lose some points.

## For Grading

Question:	1	2	3	Total
Points:	30	35	35	100
Score:				

1. (30 points) [Programming] Define a Python function, `trianglearea(base,height)`, which takes two numbers, base and height, and returns the area of a triangle with that length of base and height. Recall that the formula for the area of a triangle with base  $b$  and height  $h$  is  $b \cdot h/2$ .

Tests for this problem appear in Figure 1.

---

```
# $Id: test_trianglearea.py,v 1.1 2017/01/30 00:16:49 leavens Exp leavens $
from trianglearea import trianglearea
from math import isclose
def test_trianglearea():
    assert isclose(trianglearea(1,1),0.5)
    assert isclose(trianglearea(2,1),1.0)
    assert isclose(trianglearea(10,10),50.0)
    assert isclose(trianglearea(5,2),5.0)
```

Figure 1: Tests for `trianglearea`.

---

2. (35 points) [Programming] Define a Python function, `decreasingdigits(n)` that when given an int, `n`, where  $10 \leq n \leq 99$ , returns True just when the two digits in `n` are in strictly decreasing order. (Hint: you might want to use the `%` operator, which returns the remainder of its left argument modulo its right argument and the `//` operator, which returns the floor of the number of times its right argument divides its left argument.)

Tests for this problem appear in Figure 2.

---

```
# $Id: test_decreasingdigits.py,v 1.1 2017/01/30 00:16:49 leavens Exp leavens $
from decreasingdigits import decreasingdigits
def test_decreasingdigits():
    assert decreasingdigits(43)
    assert decreasingdigits(91)
    assert decreasingdigits(65)
    assert decreasingdigits(50)
    assert decreasingdigits(10)
    # however, note:
    assert not decreasingdigits(99)
    assert not decreasingdigits(11)
    assert not decreasingdigits(23)
    assert not decreasingdigits(39)
    assert not decreasingdigits(57)
```

Figure 2: Tests for `decreasingdigits`.

---

3. (35 points) [Programming] Define an interactive Python procedure, `digitname()`, that prompts with the string `"digit? "` (without the quotes) on `stdout`, and reads a single digit as an `int` from `stdin`. It then prints the English name for that digit on standard output. The following is a sample interaction with this program, in which the digit following the question mark (?) is a user input.

```
digit? 3  
three
```

Another interaction would be the following.

```
digit? 0  
zero
```

And the following is another interaction.

```
digit? 9  
nine
```

Your program should assume that the user gives a legal input (a digit) in response to the prompt.