

**SI@UCF Introduction to Programming in Python Test #1 Solutions**  
**7/10/2015**

1) (6 pts) Write a single python statement to print read in an integer into the variable `number` .

**`number = int(input("Please enter an integer."))`** (1 pt each part)

2) (10 pts) What are the values of the following expressions in Python? (2 pts each)

a) `5 + 3*2` **11**

b) `13%17` **13**

c) `23 - 6*(5 - 17//6)` **5**

d) `13/5 + 22%7 - 19//207` **3.6**

e) `6 + 24/(13 - 3*(5000%100 + 1234%5))` **30**

3) (12 pts) Jimmy's box for crayons has dimensions  $L \times W \times H$ . Each crayon he puts in has a height of exactly  $H$ . Thus, he stands each crayon up in the box forming rows of crayons, as if the cross section of each crayon was a square of side length  $S$ . (Imagine filling a rectangle of dimensions  $L \times W$  with squares with dimensions  $S \times S$ .) Complete the program below so that it correctly calculates the maximum number of crayons that can fit in Jimmy's box.

```
def main():

    mylen = int(input("What is the length of your box?\n"))
    mywidth = int(input("What is the width of your box?\n"))
    side = int(input("What is the diameter of each crayon?\n"))

    sqlen = mylen//side           # 4 pts
    sqwidth = mywidth//side     # 4 pts
    crayons = sqlen*sqwidth     # 3 pts
                                # print 1 pt

    print("You can fit", crayons , "crayons in the box.")

main()
```

4) (12 pts) If Jenna has an SAT score of 1900 or more AND a GPA of 3.75 or more, she earns a \$5000 scholarship. If she has an SAT score of 1900 or more OR a GPA of 3.75 or more, she earns a \$3000 scholarship, otherwise, she earns no scholarship. Complete the program below so that it prints out the value of Jenna's scholarship.

```
def main():

    sat = int(input("What is your SAT score?\n"))
    gpa = int(input("What is your GPA?\n"))

    if sat >= 1900 and gpa >= 3.75:           # 4 pts
        print("$5000 scholarship")           # 1 pt
    elif sat >= 1900 or gpa >= 3.75:         # 4 pts
        print("$3000 scholarship")           # 1 pt
    else:                                       # 1 pt
        print("No scholarship")              # 1 pt

main()
```

5) (15 pts) A tower sequence is defined by two numbers: a base and the number of terms. The first term of the sequence is the base. Each subsequent term of the sequence is the previous term squared. For example, the tower sequence with base 2 of 4 terms is 2, 4, 16, 256. Complete the program below so that it calculates the sum of a given tower sequence.

```
def main():

    numTerms = int(input("How many terms in the sequence?\n"))
    base = int(input("What is the base of the sequence?\n"))
    total = 0

    for i in range(numTerms):                 # 5 pts
        total = total + base                  # 5 pts
        base = base * base                    # 5 pts

    print("The sum of the sequence is ",total)

main()
```

6) (10 pts) What is the output of the following Python program?

```
def main():  
  
    a = 2  
    b = 7  
    for i in range(5):  
        c = b - 2*a  
        print(a,b,sep="," ,end=" : ")  
        b = c  
        a = c - a  
main()
```

**2,7 : 1,3 : 0,1 : 1,1 : -2,-1 : (Grading: 1 pt per number)**

7) (20 pts) Write a python program below using the turtle so that it creates a diamond and fills it red. Each side of the diamond should be length 100, and its internal angles should be 60°, 120°, 60° and 120°, respectively, in clockwise order. For full credit, you must use a for loop with an if statement inside of it.

```
import turtle  
  
turtle.left(60) # 2 pts  
turtle.begin_fill() # 2 pts  
turtle.fillcolor("red") # 2 pts  
for i in range(4): # 2 pts  
    turtle.forward(100) # 2 pts  
    if i%2 == 0: # 3 pts  
        turtle.left(60) # 2 pts  
    else: # 1 pt  
        turtle.left(120) # 2 pts  
turtle.end_fill() # 2 pts
```

8) (10 pts) Recall that the function `random.randint(a,b)` returns a random integer in between a and b, inclusive. Complete the program below so that it simulates rolling 1000 dice with 20 sides each, labeled 1 through 20, inclusive, and keep track of the number of times 20 was rolled, printing this out.

```
import random

def main():

    num20 = 0

    for i in range(1000):
        roll = random.randint(1,20)
        if roll == 20:
            num20 = num20 + 1

    print("You rolled 20", num20, "number of times.")

main()
```

9) (5 pts) What color are Red Hot candies? **Red Hot** (5 pts give to all)