Name: _____

With Solutions in Red

CS110 Introduction to Computing
Fall 2012 – Section 2
Exam 1

This is an open notes exam. Computers are not permitted. Your work on this exam must be your own. Answer all questions in the space provided, continuing your answer on the back of the page if necessary.

All program statements you write should be syntactically correct. Partial credit is not guaranteed with incorrect use of syntax.

Be certain to use proper indentation in any code you write. Comment your code when necessary to ensure proper understanding.

Be certain to sign the log-in sheet for the examination period, and the statement of academic honesty on the exam cover page (below).

Sign the following statement **after** you have completed the examination. Your exam will **not be graded** without your signature.

> I certify that my responses in this examination are solely the product of my own work and that I have fully abided by the Bryn Mawr College Computer Science Academic integrity policy and instructions stated above while taking this exam.
>
> Signature: _____
>
> Printed Name: _____

For Instructor Use Only:

Question #	Points	Maximum Points
1.		**10**
2.		**10**
3.		**10**
4.		**10**
5.		**10**
6.		**10**
7.		**10**
8.		**15**
9.		**20**
Total		**100**

Name: _____

1: Graphics (10 points total) Give concise, short answers to each question.

1.1. (1 pts) Write your name on EVERY page of the exam booklet and any scrap paper you use.

1.2. (2 pts) What happens when the following commands are executed?

```
from Processing import *
window(500, 500)

w, h = width(), height()
stroke(0)
line(0, 0.5*h, w, 0.5*h)
```

A horizontal black line will be drawn through the center of the entire sketch window.

1.3. (2 pts) Write a command to draw a circle of diameter 30, centered in the middle of the sketch window.

```
ellipse( 0.5*width(), 0.5*height(), 30, 30 )
```

1.4. (2 pts) The following program draws a square on a sketch. In which quadrant is the rectangle drawn?

```
from Processing import *
window(500, 500)
w, h = width(), height()
pushMatrix()
translate( 0.75*w, 0.75*h )
rotate( radians(45) )
rect(0, 0, 50, 50)
popMatrix()
```

A) Upper left
B) Upper right
C) Lower left
D) Lower right

1.5 (3 pts) Modify the program in 1.4 to also rotate the rectangle by 45 degrees after it is translated.

See modifications above.

Name: _____

2: Operators and Expressions (10 points total)

What is the value of `x` after the following expressions are executed?

2.1 (2 pts)
```
x = 3
x += 3
```

6

2.2 (2 pts)
```
x = 5
x = x // 2
```

2

2.3 (2 pts)
```
x = 5 % 2
```

1

2.4 (2 pts)
```
x = (20 <= 20)
```

True

2.5 (2 pts)
```
y = 0
x = (y > 0 or y < 0)
```

False

Name: _____

3: Conditionals (10 pts)

Write a conditional expression that would print to the screen "Too cold!" when the value of a variable T is less than 50, "Too Hot!" when T is greater than 100, and "Just Right!" otherwise.

```
if T < 50:
    print("Too Cold!")
elif T > 100:
    print("Too Hot!")
else:
    print("Just Right!")
```

4: Loops (10 pts)

Rewrite the following code using a `while` loop instead of a `for-in` loop.

```
directions = ["north", "south", "east", "west"]

for d in directions:
    print(d)
```

```
i = 0
while i < len(directions):
    print( directions[i] )
    i = i + 1
```

Name: _____

5: Functions (10 pts)

Write a function named `sum` that takes two integer number arguments, `a` and `b`, and returns the sum of all integers starting at `a` and ending at `b`, including `a` and `b`. For example, `sum(2, 5)` should return 14, which equals 2+3+4+5. You can assume that `a` and `b` will have integer values, and that `a` is strictly less than `b`.

```
def sum(a, b):
    s = 0
    n = a
    while n <= b:
        s = s + n
        n = n + 1
    return s

print( sum(2, 5) )
```

6: Variable Visibility (10 pts)

What will be printed when the following program runs?

```
def test(y):
    global x
    x = 30.0
    print(x)
    print(y)

x = 20.0

test(x)
print(x)
```

```
30.0
20.0
30.0
```

Name: _____

7: Lists (10 pts)

Complete the function named `evens()` below so that it loops over all elements in its list argument (`nums`) and prints only the even numbers it encounters.

```
def evens( nums ):
    # Define the function implementation here
    for n in nums:
        if n % 2 == 0:
            print(n)

# Init the list and pass it to the evens() function
nums = [2, 5, 7, 8, 10]
evens( nums )
```

Name: _____

8: Debugging (15 pts)

The following program is supposed to draw three blue circles on the sketch window. The x-y location of all circles doesn't change as the program runs, and is randomly reset to different values each time the mouse is pressed. The fill color of a circle is supposed to change to green whenever the mouse is over a circle. Unfortunately, my program doesn't work correctly. I'm so frustrated! Here's what doesn't work:

- First, the fill color changes too soon, before the mouse is over the circle.
- Second, when I click the button, new circles are drawn, but the old ones don't go away.

Please tell me how to fix my program. Use the back of this sheet, to explain your answer.

```
from Processing import *
window(500, 500)

Xs, Ys = [], []

# Initialize random coordinates for all three circles
def resetLocations(o, e):
    global Xs, Ys
    Xs, Ys = [], []
    for i in range(3):
        Xs.append( random(500) )
        Ys.append( random(500) )

# Draw circles at predefined random locations
def draw(o, e):
    background(255)
    for i in range(3):
        # Change fill color if mouse is over circle
        if dist( Xs[i], Ys[i], mouseX(), mouseY() ) < 2040:
            fill(0, 255, 0)
        else:
            fill(0, 0, 255)

        # Draw circle
        ellipse( Xs[i], Ys[i], 40, 40 )

# Init locations at start of program
resetLocations(None, None)

# Reset locations whenever the mouse is pressed
onMousePressed += resetLocations

# Handle onLoop timer event with draw() function
frameRate(30)
onLoop += draw
loop()
```

Name: _____

9: Classes (20 pts)

Write a class named `Ring` with four instance variables named **`x`**, **`y`**, **`d`** and **`r`**, that represent the `Ring`'s location (`x`, `y`), its diameter (`d`), and its red color value (`r`). The `Ring` constructor should initialize these instance variables using four arguments passed into it. Give the `Ring` class one method named `display()` that draws the `Ring` centered at its `x-y` location, with diameter `d`, and it's red stroke color value set to `r`. `Ring`s have no fill color.

A start of the class definition has been provided, along with a program to test the `Ring` class. You may use the back of the page if necessary.

```
from Processing import *
window(500, 500)

class Ring:
    # Class implementation goes here
    def __init__(self, x, y, d, r):
        self.x = x
        self.y = y
        self.d = d
        self.r = r

    def display(self):
        stroke(self.r, 0, 0)
        noFill()
        ellipse( self.x, self.y, self.d, self.d )

# Test the Ring class
myPrecious = Ring(100, 100, 50, 255)
myPrecious.display()
```