Quiz 1

1. Consider the following program. What will be printed if the user enters "5" for the radius? (You may use a calculator).

```
r = 0
PI = 3.14 #Close enough!

# Formula for the perimeter of a circle given its radius
C = 2 * PI * r

# Get the radius from the user
r = float(input("Enter the circle's radius:  "))

# Print the circumference
print("Circumference is ", C)
```

Answer: The program will print "Circumference is 0", because the variable `C` was assigned when `r` was still still 0.

2. What is the result of each expression?

    (a) `(5 > 4) and (3 == 5)`
        Answer: True and False→ False

    (b) `not (5 > 4)`
        Answer: not True→ False

    (c) `(5 > 4) or (3 == 5)`
        Answer: True or False→ True

    (d) `not ((5 > 4) or (3 == 5))`
        Answer: not (True or False)→ not True→False

    (e) `(True and True) and (True == False)`
        Answer: True and False→ False

    (f) `(not False) or (not True)`
        Answer: True or False→ True

3. In the following program, `i`,`j` and `k` are variables which have been previously assigned to.

```
if i < j:
  if j < k:
    i=j
  else:
    j=k
else:
  if j > k:
    j=i
  else:
    i=k
print("i =", i, " j =", j, " k =", k)
```

   What will the code print if the variables `i`,`j` and `k` have the following values?

    (a) `i=3, j=5, k=7` **Answer: `i=5,j=5,k=7`**

    (b) `i=5, j=7, k=3` **Answer: `i=5,j=3,k=3`**

    (c) `i=7, j=3, k=5` **Answer: `i=5,j=3,k=5`**

4. Are the following programs equivalent?

(a)
```
user_input = 1
while user_input % 2:
  user_input = int(input("Enter a number"))
  if user_input % 2 == 1:
    continue
  print("You entered an even number!")
```

(b)
```
while True:
  user_input = int(input("Enter a number"))
  if user_input % 2 == 0:
    print("You entered an even number!")
    break
  continue
```

Answer: In program (a): user_input starts out at 1, user_input$\%2 = 1$, so the we enter the loop. If the user enters an even number, the if block does not run, and skip to the print statement. We then proceed to check the condition again, which is now False, so we do not enter the loop again. If the user enters an odd number, we enter the if block, which, through the continue statement, moves us back to the while loop condition, which is still True, and so we loop again.

In program (b), we always enter the loop. If the user enters an even number, we enter the if block, print a message, and then jump immediately to the end of the program via the break statement. If the user enters an odd number, we skip the if block and jump down to the continue statement, which is redundant here since we were going to jump back to the while loop condition at the end of the block anyway.

Although the code is written differently, logically the two programs are the same.