In [2]:
```
#1.What will the following Python program print out?
def fred():
    print("Zap")
def jane():
    print("ABC")
jane()
fred()
jane()
```

```
ABC
Zap
ABC
```

In [3]:
```
#Solution to number 1: The progrom will print out "ZAP" for fred because that was the
#also print out "ABC" for jane because that is the string assigned to jane.
#It will look like this
ABC
ZAP
ABC
```

In [5]:
```
#2.What will the following Python program print out?
def print_lyrics():
    print("I'm a lumberjack, and I'm okay.")
    print("I sleep all night and I work all day.")
def repeat_lyrics():
    print_lyrics()
    print_lyrics()
repeat_lyrics()
```

```
I'm a lumberjack, and I'm okay.
I sleep all night and I work all day.
I'm a lumberjack, and I'm okay.
I sleep all night and I work all day.
```

In [ ]:
```
#Solution to number 2: This program will print out the string assigned to print_lyrice
#a total of two times.
#It will look like this
I'm a lumberjack, and I'm okay.
I sleep all night and I work all day.
I'm a lumberjack, and I'm okay.
I sleep all night and I work all day.
```

In [9]:
```
#3.What will the following Python program print out?
import math
def print_twice(bruce):
    print(bruce)
    print(bruce)
print_twice('Spam')
print_twice(50)
print_twice('Spam'*4)
print_twice('Spam'*20)
print_twice(math.pi)
print_twice(math.sqrt(4))
```

```
Spam
Spam
50
50
SpamSpamSpamSpam
SpamSpamSpamSpam
SpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpam
SpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpam
3.141592653589793
3.141592653589793
2.0
2.0
```

In [ ]:
```
#Solution to number 3: This program will print out the assigned varable or string assc
#In this program there are using multiplcation to increase the aditonial message befor
#and square root.
#It will look like this
Spam
Spam
50
50
SpamSpamSpamSpam
SpamSpamSpamSpam
SpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpam
SpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpamSpam
3.141592653589793
3.141592653589793
2.0
2.0
```

In [25]:
```
#4.A function object is a value you can assign to a variable or pass as an argument. F
#that takes a function object as an argument and calls it twice.
#Here's an example that uses do_twice to call a function named print_spam twice.

def do_twice(f,a):
    print("do_twice before")
    f(a)
    print("do_twice next")
    f(a)
    print("do_twice last")
def print_spam(b):
    print('spam and',"recived value =",b)

do_twice(print_spam,10)
```

```
do_twice before
spam and recived value = 10
do_twice next
spam and recived value = 10
do_twice last
```

In [26]:
```
#5. Rewrite your pay computation problem of previous lab (Lab 4, 1st exercise) by crea
#function called computepay which takes two parameters (hours and rate) and calculates
#and returns the pay value to the called place to display it.
#Sample output1:
#Enter Hours: 21
#Enter Rate: 8.25
#Pay: 173.25
#Sample output1:
#Enter Hours: 45
```

```
#Enter Rate: 8.25
#Pay: 421.25
```

In [12]:
```
#Solution for Number 5:

def computePay(hr,rate):
    print("computePay's pay being displayed here:",hr * rate)
    return (hr*rate)
flagInputError = False
try:
    hr = int(input("Enter how many hours you worked: "))
    rate = float(input("Enter how much money you make hourly: "))
except:
    flagInputError = True
    print("must enter numeric input")

if(flagInputError == False):
    grossPay = hr * rate
    if hr > 40 :
        grossPay = grossPay + 50
        print("After working",hr,"hours you will make",grossPay,"dollars with overtime
    else:
        print("After working",hr,"hours you will make",grossPay,"dollars")
print("pay is :",computePay(hr,rate))
```

```
Enter how many hours you worked: 10
Enter how much money you make hourly: 10
After working 10 hours you will make 100.0 dollars
computePay's pay being displayed here: 100.0
pay is : 100.0
```

In [31]:
```
#6. Rewrite the grade program from the previous lab (Lab 4, 2nd exercise) using a func
#computegrade that takes a score as its parameter and returns a grade as a string.
#Score Grade
#>= 0.9 A
#>= 0.8 B
#>= 0.7 C
#>= 0.6 D
#< 0.6 F
#Enter score: 0.95
#A
#Enter score: perfect
#Bad score
#Enter score: 10.0
#Bad score
#Enter score: 0.75
#C
#Enter score: 0.5
#F
#Run the program repeatedly to test the various different values for input.
```

In [11]:
```
#Solution for Number 6:
def computeGrade():
    if score > 1 or score < 0:
        return "Bad Score"
    elif score >= 0.9 :
        grade = "A"
    elif score >= 0.8 :
        return "B"
```

```
    elif score >= 0.7 :
        return "C"
    elif score >= 0.6 :
        return "D"
    elif score < 0.6 :
        return "F"
    return grade

flagInputError = False
try:
    score = float(input("enter your number here: "))
except:
    flagInputError = True

if(flagInputError == False):

    grade = computeGrade()
    print(grade)
else:
     print("must use numeric input")
```

```
enter your number here: uuu
must use numeric input
```

In [ ]: