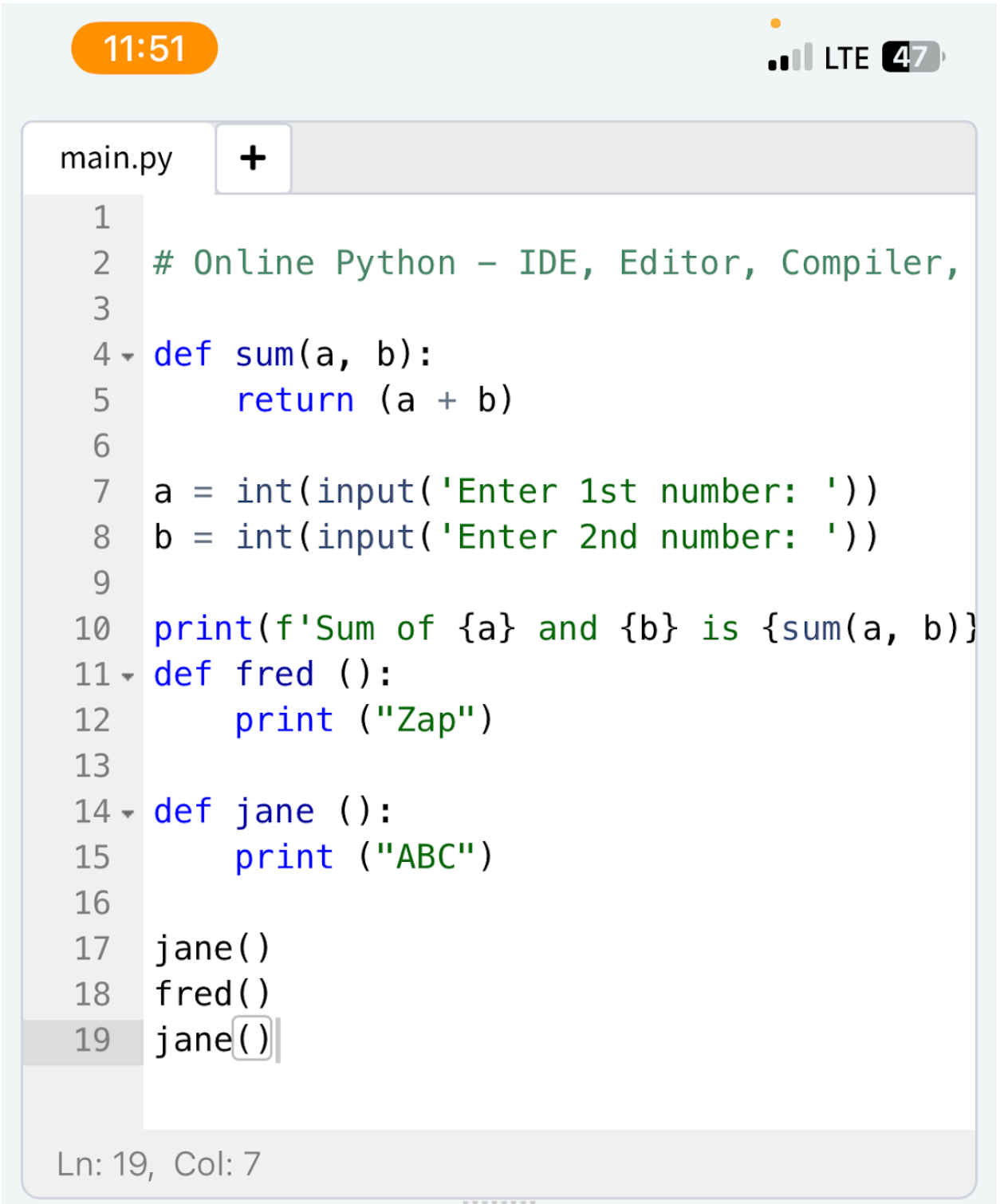
Borja, Seth N.
CPE 310 - Fundamentals of Data Science
Exercise 1.2 Functions, Loops, and Iterations

Part 1
Exercise 1
D.
Exercise 2
D.

```
def fred ():
    print ("Zap")

def jane ():
    print ("ABC")

jane()
fred()
jane()

def computepay(Hours, Rate):
    if hours > 40:
    regular_pay = 40 * Rate
    overtime_pay = (Hours - 40) * (Rate * 1.5)
    Total_pay = Regular_pay + Overtime_pay
    else:
        Total_pay = Hours * Rate
        return Total_pay
```


11:51
LTE 47
main.py
1
2 # Online Python - IDE, Editor, Compiler,
3
4 def sum(a, b):
5 return (a + b)
6
7 a = int(input('Enter 1st number: '))
8 b = int(input('Enter 2nd number: '))
9
10 print(f'Sum of {a} and {b} is {sum(a, b)}
11 def fred ():
12 print ("Zap")
13
14 def jane ():
15 print ("ABC")
16
17 jane()
18 fred()
19 jane()
Ln: 19, Col: 7
Run
Share
Command Line Arguments
Enter 1st number: 10
Enter 2nd number: 20
Sum of 10 and 20 is 30
ABC
Zap

Exercise 3

```
def computepay(hours, rate):
    if hours > 40:
        regular_pay = 40 * rate
        overtime_pay = (hours - 40) * (rate * 1.5)
        total_pay = regular_pay + overtime_pay
    else:
        total_pay = hours * rate
    return total_pay


hrs = 45
rte = 10
pay = computepay(hrs, rte)
print("Pay:", pay)
```

main.py | Untitled2.py | +

```
1  def computepay(hours, rate):
2      if hours > 40:
3          regular_pay = 40 * rate
4          overtime_pay = (hours - 40) * (ra
5          total_pay = regular_pay + overtim
6      else:
7          total_pay = hours * rate
8      return total_pay
9
10
11 hrs = 45
12 rte = 10
13 pay = computepay(hrs, rte)
14 print("Pay:", pay)
15
```

Ln: 10, Col: 1

Run | Share | Command Line Arguments

```
Pay: 475.0
```

Exercise 4

```
def computegrade(score):
    try:
        score = float(score)
    except:
        return "Bad score"

    if score < 0.0 or score > 1.0:
        return "Bad score"
    elif score >= 0.9:
        return "A"
    elif score >= 0.8:
        return "B"
    elif score >= 0.7:
        return "C"
    elif score >= 0.6:
        return "D"
    else:
        return "F"



while True:
    s = input("Enter score: ")
    if s == "done":
        break
    print(computegrade(s))
```

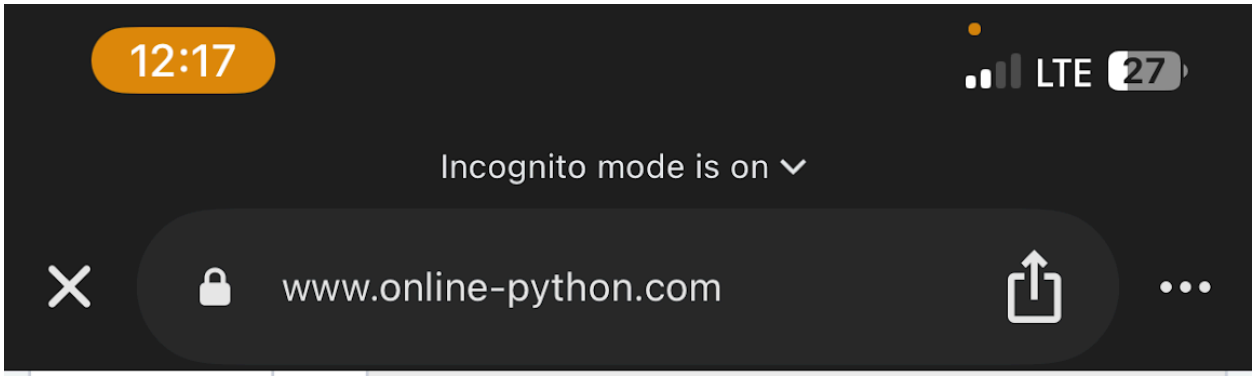
main.py +

```
    except:
        return "Bad score"

    if score < 0.0 or score > 1.0:
        return "Bad score"
    elif score >= 0.9:
        return "A"
    elif score >= 0.8:
        return "B"
    elif score >= 0.7:
        return "C"
    elif score >= 0.6:
        return "D"
    else:
        return "F"


while True:
    s = input("Enter score: ")
    if s == "done":
        break
    print(computegrade(s))
```

Ln: 21, Col: 1

Stop | Share | Command Line Arguments

```
Enter score: 0.9
```



main.py

```
1
2  def computegrade(score):
3      try:
4          score = float(score)
5      except:
6          return "Bad score"
7
8      if score < 0.0 or score > 1.0:
9          return "Bad score"
10     elif score >= 0.9:
11         return "A"
12     elif score >= 0.8:
13         return "B"
14     elif score >= 0.7:
15         return "C"
16     elif score >= 0.6:
17         return "D"
```

Ln: 21, Col: 1

Stop | Share | Command Line Arguments

```
Enter score: 0.9
A
Enter score: 0.8
B
Enter score: 0.7
```

Part 2
Exercise 1

```
total = 0
count = 0

while True:
    num = input("Enter a number: ")
    if num == "done":
        break
    try:
        value = int(num)
    except:
        print("Invalid input")
        continue
    total += value
    count += 1

if count > 0:
    average = total / count
    print(total, count, average)
else:
    print("No numbers entered")
```

main.py Untitled2.py +

```
total = 0
count = 0

while True:
    num = input("Enter a number: ")
    if num == "done":
        break
    try:
        value = int(num)
    except:
        print("Invalid input")
        continue
    total += value
    count += 1

if count > 0:
    average = total / count
    print(total, count, average)
else:
```

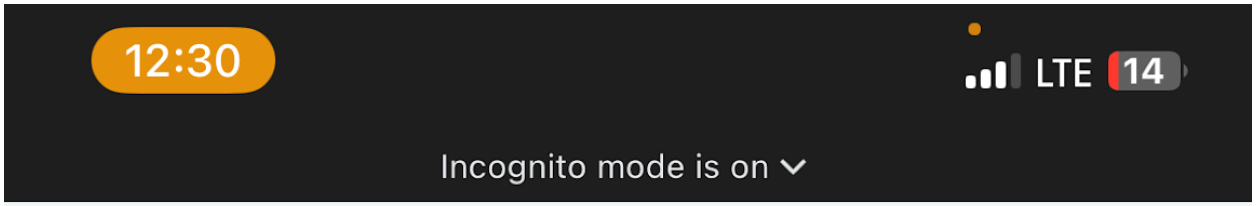
Ln: 16, Col: 14

Run Share Command Line Arguments

```
Enter a number: 4
Enter a number: 5
```

Exercise 2

```
numbers = []
while True:
    user_input = input("Enter a number: ")

    if user_input.lower() == "done":
        break

    try:
        num = int(user_input)
        numbers.append(num)
    except ValueError:
        print("Invalid input")
if numbers:  # only if the list is not empty
    print("Maximum:", max(numbers))
    print("Minimum:", min(numbers))
else:
    print("No valid numbers were entered.")
```



main.py Untitled2.py Untitled3.py +

```
1  umbers = []
2
3  hile True:
4      user_input = input("Enter a number: ")
5
6      if user_input.lower() == "done":
7          break
8
9      try:
10         num = int(user_input)
11         numbers.append(num)
12     except ValueError:
13         print("Invalid input")
14
15 f numbers:  # only if the list is not empt
```

Ln: 19, Col: 44

Run Share Command Line Arguments

```
Enter a number: 5
Enter a number: 2
Enter a number: 9
Enter a number: bad data
Invalid input
```