list\_comprehensions:

*"""  
CP1404/CP5632 Practical  
List comprehensions  
"""*names = ["Bob", "Angel", "Jimi", "Alan", "Ada"]  
full\_names = ["Bob Martin", "Angel Harlem", "Jimi Hendrix", "Alan Turing",  
 "Ada Lovelace"]  
  
# for loop that creates a new list containing the first letter of each name  
first\_initials = []  
for name in names:  
 first\_initials.append(name[0])  
print(first\_initials)  
  
# list comprehension that does the same thing as the loop above  
first\_initials = [name[0] for name in names]  
print(first\_initials)  
  
# list comprehension that creates a list containing the initials  
# splits each name and adds the first letters of each part to a string  
full\_initials = [name.split()[0][0] + name.split()[1][0] for name in  
 full\_names]  
print(full\_initials)  
  
# one more example, using filtering to select only the names that start with A  
a\_names = [name for name in names if name.startswith('A')]  
print(a\_names)  
  
# *TODO: use a list comprehension to create a list of all of the full\_names*# in lowercase format  
lowercase\_full\_names = [name.lower() for name in full\_names]  
print(a\_names)  
  
almost\_numbers = ['0', '10', '21', '3', '-7', '88', '9']  
# *TODO: use a list comprehension to create a list of integers*# from the above list of strings  
numbers = [int(number) for number in almost\_numbers]  
  
# *TODO: use a list comprehension to create a list of only the numbers that are*# greater than 9 from the numbers (not strings) you just created  
greater\_than\_nine = [number for number in numbers if number > 9]

list\_exercises:

def main():  
 num1 = int(input("Number: "))  
 num2 = int(input("Number: "))  
 num3 = int(input("Number: "))  
 num4 = int(input("Number: "))  
 num5 = int(input("Number: "))  
 num\_list = (num1, num2, num3, num4, num5)  
 print("The first number is:", num1)  
 print("The last number is:", num5)  
 print("The smallest number is:", min(num\_list))  
 print("The largest number is:", max(num\_list))  
 print("the average of the number is:", (sum(num\_list)) / 5)  
  
 sernames = ['jimbo', 'giltson98', 'derekf', 'WhatSup', 'NicolEye', 'swei45', 'BaseInterpreterInterface',  
 'BaseStdIn',  
 'Command', 'ExecState', 'InteractiveConsole', 'InterpreterInterface', 'StartServer', 'bob']  
 username = input("What is your username?: ")  
 if username in sernames:  
 print("Access granted")  
 else:  
 print("Access denied")  
  
  
main()

lists\_warmup:

numbers = [3, 1, 4, 1, 5, 9, 2]  
"""  
What values do these have?  
numbers[0]  
numbers[-1]  
numbers[3]  
numbers[:-1]  
numbers[3:4]  
5 in numbers  
7 in numbers  
"3" in numbers  
numbers + [6, 5, 3]  
  
numbers[0]: 3  
numbers[-1]: 2  
numbers[3]: 1  
numbers[:-1]: error  
numbers[3:4]: 1, 5  
5 in numbers: numbers [4]  
7 in numbers: ?  
"3" in numbers: numbers[0]  
numbers + [6, 5, 3]: [3, 1, 4, 1, 5, 9, 2, 6, 5, 3]  
  
"""  
# question 1:  
numbers[0] = "ten"  
# question 2:  
numbers[-1] = "1"  
# question 3:  
numbers[2:]  
# question 4:  
9 in numbers  
print(numbers)

quick\_picks:

import random  
  
  
def main():  
 line\_num = 6  
 minimum = 1  
 maximum = 45  
  
 quick\_picks = int(input("How many quick picks do you want to generate?: "))  
  
 for picks in range(quick\_picks):  
 quick\_pick = []  
 for lines in range(line\_num):  
 random\_number = random.randint(minimum, maximum)  
 while random\_number in quick\_pick:  
 random\_number = random.randint(minimum, maximum)  
 quick\_pick.append(random\_number)  
 quick\_pick.sort()  
 print(" ".join("{:2}".format(number) for number in quick\_pick))  
  
  
main()

subject\_reader:

*"""  
CP1404/CP5632 Practical  
Data file -> lists program  
"""*FILENAME = "subject\_data.txt"  
  
  
def main():  
 data = get\_data()  
 subject\_info(data)  
  
  
def get\_data():  
 *"""Read data from file formatted like: subject,lecturer,number of students."""* data = []  
 input\_file = open(FILENAME)  
 for line in input\_file:  
 print(line) # See what a line looks like  
 print(repr(line)) # See what a line really looks like  
 line = line.strip() # Remove the \n  
 parts = line.split(',') # Separate the data into its parts  
 print(parts) # See what the parts look like (notice the integer is a string)  
 parts[2] = int(parts[2]) # Make the number an integer (ignore PyCharm's warning)  
 print(parts) # See if that worked  
 print("----------")  
 data.append(parts)  
 input\_file.close()  
 return data  
  
  
def subject\_info(data):  
 for subject in data:  
 print("{} is taught by {:12} and has {:3} students".format(\*subject))  
  
  
main()

total\_income:

*"""  
CP1404/CP5632 Practical  
Starter code for cumulative total income program  
"""*def main():  
 *"""Display income report for incomes over a given number of months."""* incomes, number\_of\_months = income\_per\_months()  
  
 print("\nIncome Report\n-------------")  
 total = 0  
 for month in range(1, number\_of\_months + 1):  
 income = incomes[month - 1]  
 total += income  
 print("Month {:2} - Income: ${:10.2f} Total: ${:10.2f}".format(month, income, total))  
  
  
def income\_per\_months():  
 incomes = []  
 number\_of\_months = int(input("How many months? "))  
 for month in range(1, number\_of\_months + 1):  
 income = float(input("Enter income for month {}:".format(month)))  
 incomes.append(income)  
 return incomes, number\_of\_months  
  
  
main()