Assignment 1

Nathaniel Martinez
CAAP-CS

July 12, 2019

Part 1

1 Hello World

In order to print out Hello World, I used the following code:

```
print("Hello World")
```

2 Converter

In order to convert from Fahrenheit to Celsius, I used the formula (FTemp - 32)*5/9= CTemp. From there I used a for loop to print the temperature 5 times.

```
count =0
FTemp = eval(input("What is the temperature in Fahrenheit?"))
CTemp = (FTemp-32)*(5/9)
for count in range(5):
    print(CTemp)
```

3 Unit Conversion

In order to convert from yards to feet, you must multiply the yards by 3 since there are 3 feet in a yard:

```
print("This is a program that converts from yards to feet!")
yards = eval(input("How many yards do you want to convert to feet?"))
feet = yards*3
print("There are ",feet," feet in ",yards," yards.")
```

4 Slope

To find out the sum of a certain amount of numbers, you will need 3 variables. First, 1 variable will take in the amount of numbers the user wants to sum. The second variable will keep track of the sum as each number is added. Finally, the third variable will be the number the user wants to add. A for loop is needed that will run according to the amount of numbers needed to be summed. Nested inside the for loop is a sum function and a question asking the user to input a number.

```
sum =0
nums = eval(input("How many numbers do you want to sum?"))
for count in range(nums):
    value = eval(input("Please enter your number:"))
    sum += value
print("The sum of the numbers are:",sum)
```

5 Fibonacci Sequence

Calculating the nth Fibonacci Sequence requires the use of a for loop, a temporary variable and a bit of intuition. If the user inputs either one or two, then the program will automatically output one. If not, the program with use a for loop to calculate the Fibonacci number using temporary variables in order to store the values of the previous 2 numbers.

```
pos =eval(input("This program will calculate the nth Fibonacci number. Please
input the nth fibonacci number this program will calculate to:"))
if pos == 1 or pos == 2:
    print(1)
first=1
second=1
for count in range(pos-2):
    temp = second
    second = first + second
    first = temp
print("The",pos,"th Fibonacci number is ",second)
```

Part 2 - Change

In order to calculate the least amount of coins needed in order to give change for a specific value, I used a series of if statements and variables corresponding to the value of specific coins. I first used division by the value of the coins (change/0.25(the value of a quarter)). If the value of the division is less than 1, then the value cannot be extracted (since it is less than the value). If it was greater than 1, the program was kicked into a while loop which

broke when the value of the change was less than the value of the coin. I continued this for every coin (quarters, dimes, nickels, and pennies) until I was left with 0.00 left in change. This worked initially, however I soon encountered a bug. For some reason, my computer would subtract a small fraction of the change everytime, ultimately making it less than a penny. As a result, I had to implement another if statement in order ot work around the bug. I established a tolerance of 0.0001. I added 0.0001 to the change everytime it was within 0.0001 of the next coin (change had a value of 0.0099999, I added 0.0001 in order to subtract an extra penny). The code is below:

```
change =float(input("How much change do you have?"))
   qcount =0
   dcount =0
   ncount =0
   pcount =0
   coins =0
   if change/0.25 ¿= 1:
      while change¿=0.25:
         change-=0.25
         qcount+=1
   if (change+0.0001)/0.25¿=1:
      change-=0.25
      qcount+=1
   if change/0.10 ¿= 1:
      while change¿=0.10:
         change-=0.10
         dcount+=1
   if (change+0.0001)/0.2¿=1:
      change-=0.2
      dcount+=1
   if change/0.05 ¿= 1:
      while change¿=0.05:
         change-=0.05
         ncount+=1
   if (change+0.0001)/0.05¿=1:
      change-=0.05
      qcount+=1
   if change/0.01 ¿= 1:
      while change¿=0.01:
         change-=0.01
         pcount+=1
   if (change+0.0001)/0.01¿=1:
      change-=0.01
      qcount+=1
   coins+=qcount
   coins+=dcount
```

```
coins+=ncount
coins+=pcount
if change¡0.00001
   change=0
print(coins)
print(change)
```

Part 4 - Questions

1. What is a greedy algorithm - you can look up definitions online but final response should be in your own words.

A greedy algorithm is an algorithm designed to be the most efficient algorithm possible at each step. However, there are several limitations to a greedy algorithm. The algorithm, for example, can only take into consideration the information it has at its current step. As a result, the algorithm comes to inaccurate solutions when it thinks a certain path is optimal when in reality it is not.

2. How else could you solve the above problem?

You could solve the change problem by continuously breaking down the values more and more until you eventually hit 0 cents left. This program will leave you with the least amount of coins necessary to create any value.

3. What other problems are greedy algorithms successfully used for?

A greedy algorithm can be used to solve the travelling salesman problem, used in order to find the least amount of distance between a list of cities and their positions. A greedy algorithm would compute the shortest path between the list of cities.