

# Assignment 1

*This is Spencer Hoffman's Homework 1 LATEX Document.*

## Hello World

**Example 1.** *Hello World!* is a basic program that is usually the first one anybody creates when learning a new programming language. This is how I created my version of this classic program.

```
print("Hello World")
username = input("What's your name? ")
print("Hello", username)
color = input("What is your favorite color?")
print("Wow!", color, "! That's my favorite color too!")
sadness = input("Why are you always so sad?")
print("Understandable. Have a nice day.")
attractive = input("What Kind of Person are you?")
print("Interesting")
alcoholic = input("Are you an alcoholic?")
print("You might want to change that")
final = input("Would you like to know what I think of you?")
if final == "yes":
    print("You are", username, ", an", attractive, "person whose favorite color is", color,
          ". You are currently sad because", sadness, ".")
    if alcoholic == "yes": #If you chose Alcoholic=yes, then this will play
        print("You are also an alcoholic running away from your problems")
    else: #If you don't want to hear what the program thinks of you, this plays
        print("Goodbye then")
```

---

**Explanation.** *Here I added some of my own flavor, attempting to complete the Challenge.*

---

Learning outcomes:  
Author(s): Spencer Hoffman

## Converter

**Example 2.** *This is a simple converter that will take an input in fahrenheit and convert it into celcius.*

```
print("Hello")
inittemp = eval(input("Please Enter a Temperature in Fahrenheit: "))
celcius = (init_temp - 32) * 5.0/9.0
finaltemp = "Celcius =" + str(celcius) + " C" #Conversion Function
for x in range(5): #Print the total 5 times
    print("Your temperature is", final_temp)
```

**Explanation.** *This program begins by making the operator enter a value for the variable "inittemp".*

*This variable is then inputted directly into the function finaltemp, utilizing the Celcius variable (which houses the function for converting the fahrenheit number to celcius). Finally, in accordance with the instructions I made the program print the answer five times in a row.*

## Unit Conversion

**Example 3.** *This is a simple converter that will intake a measurement in meters and output a measurement in kilometers*

```
print("Hello, I am a program that will convert any number of meters to kilometers for you")
metersvalue= eval(input("Please Enter a length in Meters: "))
convert= metersvalue/1000
if metersvalue <0: #Error message if -meters are entered
    print("ERROR: Next time Please enter a positive number")
else: #Print answer
    print ("Your Answer is: ",convert, "km")
```

**Explanation.** *Here we have a very simple function that takes user input and logs it as "metersvalue". From there it is checked for errors (aka. if the operator entered a negative number the program will terminate and tell the operator to enter a positive number and try again). If the input checks out, the program will spit out the correct measurement.*

## Slope

**Example 4.** *This program is rather simple, taking a series of numbers entered by the user and adding them all together.*

```

    print ("Hello! This program will sum a series of numbers for the user")
    numlist= int(input("Please enter the number of numbers you wish to be summed: ")) #this is t
    numnumlist= [0] #this is the list num is put into
    if numlist <1: #Error message if -valules are entered
        print("ERROR: Next time Please enter a positive value")
    for x in range(numlist):
        nums= int(input("Enter the next number: ")) #these are the numbers entered into the num
        numnumlist.append(nums) #enters nums into numnumlist
    print (sum(numnumlist))

```

**Explanation.** First off, the user must enter in the number of values they wish to enter into the program. If it passes inspection, the user will be asked to input the numbers one at a time, with the for loop insterting them into the set "numnumlist". Finally, the program will sum all of the numbers located in numnumlist and print the anser for the user.

## Fibonacci Sequence

**Example 5.** This is a more complicated program than those that came before. This program has within it a coded Fibonacci sequence, which is a sequence of values, each comprised of the sum of the two that came before.

```

    print("Hello! Welcome to your fibonacci sequence.")
    n = int(input("Please input the number of terms in your fibonacci sequence (make sure it is
    n_real = n-2
    f_list = [1, 1]
    if n < 1:#terminates program if entered value is less than 1
        print("ERROR: Next time, Please Choose a Number greater than or equal to 1")
    elif n <3: #if n is less than 3 but 1 or greater the answer is always 1
        print(1)
    else:#If the previous conditions are not transgressed, evaluate fibonacci
        for x in range(n_real):
            nums = f_list[-2] + f_list[-1] #adds the last and second last numbers added to the l
            f_list.append(nums) #adds the above sum to the list f_list
        print ("The total value of your fibonacci sequence is: ",f_list[-1])

```

**Explanation.** Here the Fibonacci sequence is coded by beginning the list with 2 terms hardcoded: those being 1,1. From there the function has to adapt to the numbers hardcoded into the system, hence " $n_{real} = n - 2$ ".

This code will also check for negative values on the scale of the fibonacci sequence and terminate if such a term is entered. If 1 or 2 is entered the program will output a 1 in accordance with the two hardcoded values. Otherwise, the function will run as intended and output the correct Fibonacci sequence value for the user.

## Cash Register

**Example 6.** *This cash register is coded utilizing GREEDY ALGORITHMS. The cash register will output the most efficient combination of coins to fulfill any change requirement.*

```

    print("Hello! This is your friendly neighborhood cash register. You want coins, we g
changevalue = float(input("Enter the Total Change of Transaction: "))
if changevalue < .01: #Error message if -change is entered
    print("ERROR: Next time Please enter a positive change value")
quartercounter = 0 #Counts the quarters
dimecounter = 0 #Counts the dimes
nickelcounter = 0 #Counts the nickels
pennycounter = 0 #Counts the pennies
while changevalue > .24: #Loop for Quarters
    changevalue -= .25 #Takes one quarter's worth of money from the cash owed
    quartercounter += 1 #adds one quarter to the total to be given to customer
while changevalue > .9: #Loop for Dimes
    changevalue -= .10
    dimecounter += 1
while changevalue > .04: #Loop for Nickels
    changevalue -= .05
    nickelcounter += 1
while changevalue > .00: #Loop for Pennies
    changevalue -= .01
    pennycounter += 1
print ("Your change is", quartercounter, "quarters", dimecounter, "dimes", nickelcounter, "r

```

**Example 7.** *This cash register will check for - change values and terminate the program if a discrepancy is detected. This register operates by simulating the cashier. For instance, with the quarters as long as there is more than 24 cents left in the change value the program will loop, taking 25 cents away from the change value and updating the number of quarters to be dispensed (quartercounter) every time. This same process is repeated with dimes, nickels, and pennies.*