Disclaimer: These notes may be distributed outside this class only with the permission of the Instructor.

3.1 Draw a Calendar with Turtle

Listing 1: Turtle Calendar Example

```python
import turtle

t = turtle.Turtle()

def line(x1, y1, x2, y2):
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)

def table():
    for i in range(7):
        line(-200, 200-i*40, 200-i*40)
    for i in range(8):
        line(-200+i*60, 200, -200+i*60, -40)

def tex():
    t.up()
    t.goto(-20, 220)
    t.color('blue')
    t.write('January 2014', False, 'center', 'Arial, 24')
    t.goto(-170, 170)
    t.color('red')
    t.write('S', False, 'center', 'Arial, 24')
    t.goto(-110, 170)
    t.color('red')
    t.write('M', False, 'center', 'Arial, 24')
    t.goto(-50, 170)
    t.color('red')
    t.write('T', False, 'center', 'Arial, 24')
    t.goto(70, 170)
    t.color('red')
    t.write('W', False, 'center', 'Arial, 24')
```

3-1
```python
def main():
    t.speed(10)
    table()
    tex()
    turtle.exitonclick()
main()
```

---

**Figure 3.1:** Draw a calendar with Turtle
3.2 Calculate PI using Monte Carlo Method

Consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is π/4, the value of π can be approximated using a Monte Carlo method.

- Draw a square on the ground, then inscribe a circle within it.
- Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.
- Count the number of objects inside the circle and the total number of objects.
- The ratio of the two counts is an estimate of the ratio of the two areas, which is π/4. Multiply the result by 4 to estimate π.

Listing 2: Calculate PI by simulation

```python
import random
import math
import turtle

def line(t, x1, y1, x2, y2):
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)

def showMontePi(numDarts):
    wn = turtle.Screen()
    t = turtle.Turtle()
    t.speed(10)
    wn.setworldcoordinates(-2,-2,2,2)
    line(t,-1,0,1,0)
    line(t,0,1,0,-1)
    circle = 0
    t.up()
    for i in range(numDarts):
        x = random.random()
        y = random.random()
        d = (math.sqrt(x**2 + y**2))
        t.goto(x,y)
        if(d <= 1):
            circle += 1
            t.color('red')
        else:
            t.color('blue')
            t.dot()
    pi = circle / numDarts * 4
    t.goto(0.2,0.5)
    t.write(pi, 'center', font=('Arial', 20, 'normal'))
    turtle.exitonclick()
```
3.3 Strings

Listing 3: Iterate each character in a string

```python
s = "hello world"
for c in s:
    print(c)
```

3.3.1 Count the frequency of letters

In this example, we count the frequencies of letters [a-z] in the book “War and Peace” by Leo Tolstoy, and draw a bar graph based on the frequency.

Listing 4: Letter Frequency Example

```python
import turtle
def letterFreq(filename):
    freq = [0 for i in range(128)]
    instr = open(filename,"r")
    count = 0
    for s in instr:
        s = s.lower()
        #print(s)
```
for c in s:
    freq[ord(c)] += 1
count += 1
print(count)
def drawRect(t, left, top, right, bottom):
    t.up()
    t.goto(left, top)
    t.down()
    t.pensize(10)
    t.goto(right, bottom)
def drawBar(fr, t):
    t.up()
    x = -250
    y = 0
    for i in range(97, 97 + 26):
        y = fr[i] / 1000
        drawRect(t, x, y, x, 0)
        t.up()
        t.goto(x, -40)
        t.write(chr(i), 'left', font=('Arial', 20, 'normal'))
        x += 20

def main():
    t = turtle.Turtle()
    #change the path on your computer
    file = '/Users/anwar/Desktop/war_peace.txt'
    fr = letterFreq(file)
    #for i in range(32, 128):
    #    if(i >= 97 and i <= 97 + 26):
    #        print(chr(i), end='	')
    #        print(fr[i])
    drawBar(fr, t)
    turtle.exitonclick()
3.3.2 Count the frequency of words

In this example, we count the frequencies of words in the book "War and Peace" by Leo Tolstoy, and display the most frequent words.

Listing 5: Word Frequency Example

```python
import re
def letterFreq(filename):
    freq=dict()
    #print(freq)
    instr = open(filename,"r")
    count = 0
    for s in instr:
        #print(s)
        s = s.lower()
        #words = s.split()
        words = re.sub(r'[?.!/\%\*"\,\-'', \s.]', '', s).split()
        for word in words:
            if freq.get(word) is None:
                freq[word] = 1
            else:
                freq[word] += 1
        count +=1
    instr.close()
    #print(freq)
```

Figure 3.3: Letter frequency
3.3.3 Anagrams

An anagram is a type of word play, the result of rearranging the letters of a word or phrase to produce a new word or phrase, using all the original letters exactly once; for example Doctor Who can be rearranged into Torchwood. This program receives an input from user, and lists all words from a dictionary that can be constructed using the user the letters of the user input. For example: “Lord Voldemort” can be constructed from “Tom Marvolo Riddle”.

Listing 7: Word Frequency Output

```python
def contains(user_input, word):
    for c in word:
        pos = user_input.find(c)
        if (pos == -1):
            return False
        else:
            word = word[:pos] + word[pos+1:]
    return True
def readWord(file, user_input):
    fin = open(file, "r")
    for word in fin:
        word = word.strip()
        if (len(word) >= 3 and contains(user_input, word)):
            print(word)
def main():
```
filename = "/Users/anwar/Desktop/wordlist.txt" #dictionary file
user_input = input("Enter a word: ")
readWord(filename, user_input)
main()