15-112 Fundamentals of Programming

Lecture 2 – Sequence and Functions

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Course ground rules

- Come to class and be on time
- No private conversations
- No cell phones/Ipads/Laptops/etc. during class.
- Do not use computers unless asked
- Bring a supply of paper and pens/pencils
- Do the readings before class and be prepared
- We start at 1:30pm. No one allowed in class after that

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Announcements

- First assignment has been posted. Due date is Tuesday Sept 1, at 10:00pm.
- Grace days
- TA meetings

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Finishing up from last time

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Getting Help from CA

- When you go to the CA, you should be able to answer three questions
  - What is the homework problem?
  - How are you solving the problem?
  - What is your code doing?
  - How have you tested it?

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Homework and online resources

Sharing….

- Concepts and background facts are OK
- Answers to problems/source code are not.
- If you're not sure, *ask an instructor before you use the resource*

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Alice and Bob

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Alice and Bob

Bob: *Hey Alice, let me see your answers to the homework.*

Alice: *Here you go!*

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Alice and Bob

Alice: *Hey Bob, I see you're having trouble with your program, let me see if I can help you find your bug*

Bob: *Oh thanks!*

Alice and Bob

Alice: *Hey Bob, I didn't quite understand chapter 3. Can we spend some time studying together for the quiz?*

Bob: *Definitely!*

Alice and Bob

Bob: *Gee Alice, I've finished the homework and I know you did too, did you want to compare answers to check our work?*

Alice: *Sounds good to me!*

Alice and Bob

Alice: *Hey Bob, you're a sophomore so you're not in my class, can you help me figure out how to do this homework?*

Bob: *Let's talk about it!*

Alice and Bob

Bob: *Hey Alice, you're a sophomore so you're not in my class, can you help me figure out how to do this homework?*

Alice: *Let's take a look at what you've written!*

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Alice and Bob

Bob: *Hmm...this homework problem requires me to know the average velocity of a bird to be able to solve the problem. I'll go look it up on Wikipedia.*

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Alice and Bob

Alice: *I found this nifty program online which almost solves one of the homework problems. I'm not going to copy it, but I'll use it as a reference when writing my own solution*

General Advice

- Leave yourself time to get help if you need it
  - Start homework early!
- Don't be afraid to get help!
  - Just be sure it's from an appropriate source
- Protect yourself
  - Don't leave yourself logged in to cluster machines
  - Don't leave printouts of your homework around

What are algorithms

- Sequence of instructions that solve a particular problem
  - So Sequence is important
  - How would you write a sequence of instructions to bake a cake?

Printing in python

- You can use the print statement to display a message on the screen

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

- How would you print a recipe on the screen?

Working with sequences

- Let's work on writing sequential instructions to draw pictures
  - If you could draw a line using the command forward and left, how would you draw a square?

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Introducing Turtle

- What is turtle?

  - Turtle is like a drawing board
  - A python **predefined** module
  - You can create a turtle and move it around
  - We need to **import** turtle!

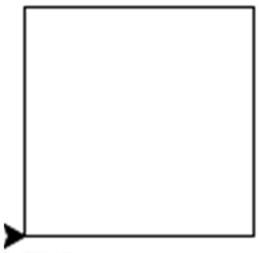
جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Turtle cheatsheet!

- **from turtle import ***
  - Call the turtle module/package with all its functions
- **forward (distance in cm)**
  - Moves the turtle forward *distance*, drawing a line behind the turtle
- **backward(distance in cm)**
  - Moves the turtle backward *distance,* drawing a line behind the turtle
- **right (angle degrees)**
  - Turns the turtle right by *angle*
- **left (angle degrees)**
  - Turns the turtle left by *angle*
- **penup()**
  - Stop all drawing until pendown is called
- **pendown()**
  - Resume drawing after a call to penup()
- **color (*color*)**
  - Change the turtle's current color
- **bye()**
  - Close turtle
- **done()**
  - Must be the last statement in a turtle graphics program

Let's play with turtle!

- **Problem** : draw a square



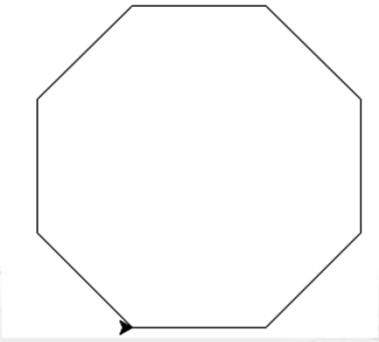
جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Square Solution

```
from turtle import *

forward(200)
left(90)
forward(200)
left(90)
forward(200)
left(90)
forward(200)
left(90)
```


It gets complicated

- **Problem**: draw an octagon



Octagon Solution

```
from turtle import *

forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
forward(200)
left(45)
```


Introduction to a loops

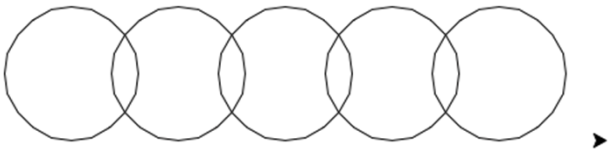
- Octagon again

```
from turtle import *

for n in range(8):
    forward(200)
    left(45)
```

→ Much better

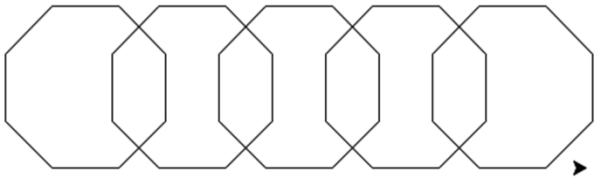
Introduction to loops

- **Problem**: draw 5 circles that overlap each other



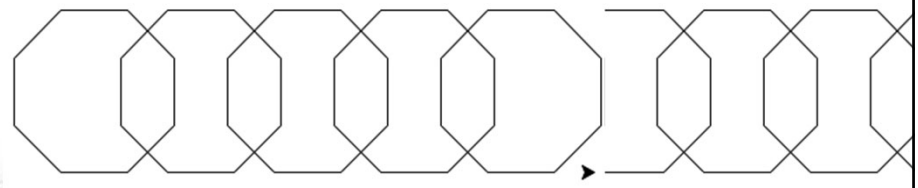
Introduction to loops

- **Problem**: draw 5 octagons that overlap each



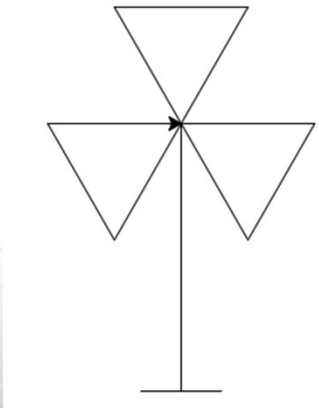
Introduction to loops

- **Problem**: draw 40 octagons that overlap each other



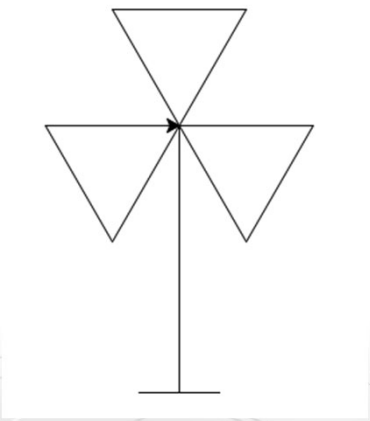
Introduction to functions

Problem: Draw a windmill



Task Decomposition

- Draw the Base
- Draw the sails



Draw Base

```
def drawBase():
    forward(100)
    right(180)
    forward(50)
    right(90)
    forward(250)
    right(90)


drawBase()
```

Draw Sails

- Draw three triangles

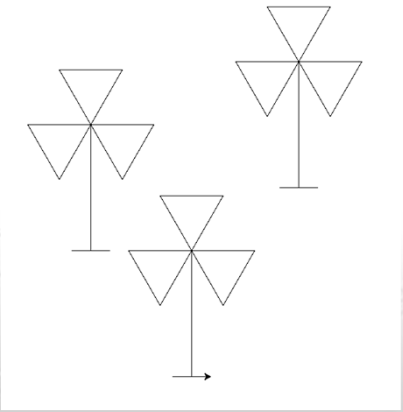
```
def triangle():
    for n in range(3):
        forward(100)
        right(120)


for n in range(3):
    triangle()
    left(120)
```


Draw the windmill

```
from turtle import *

def drawBase():
    forward(100)
    right(180)
    forward(50)
    right(90)
    forward(250)
    right(90)

def triangle():
    for n in range(3):
        forward(100)
        right(120)


drawBase()
for n in range(3):
    triangle()
    left(120)
```

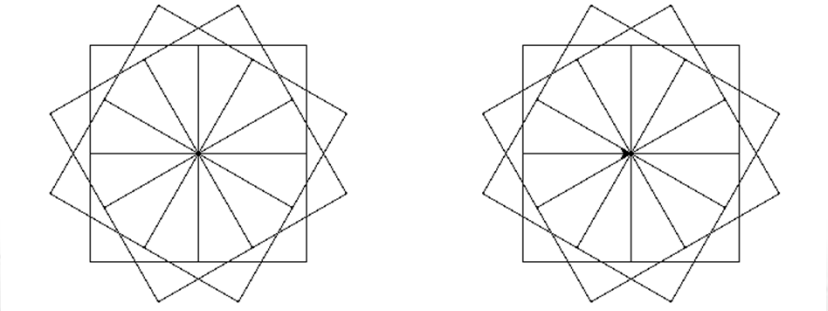
More decomposition

- What if we want to draw 3 windmills!



Introduction to functions

- Draw 2 flowers as shown in this figure



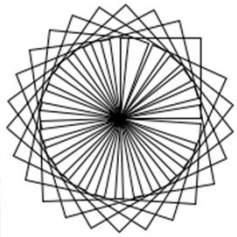
Introduction to functions

- Draw 1 flower using squares:

```
from turtle import *

def square():
    for n in range(4):
        forward(50)
        left(90)

def flower():
    for k in range(12):
        square()
        left(30)
```


Introduction to parameters

- Draw 1 flower using squares and defining the number of petals and their sizes



Introduction to parameters

- Draw 1 flower using squares:

```
from turtle import *
def square(length):
    for n in range (4):
            forward(length)
            left(90)


def flower(nbPetals, petalSize):
    for k in range (nbPetals):
            square(petalSize)
            left(360/nbPetals)
```