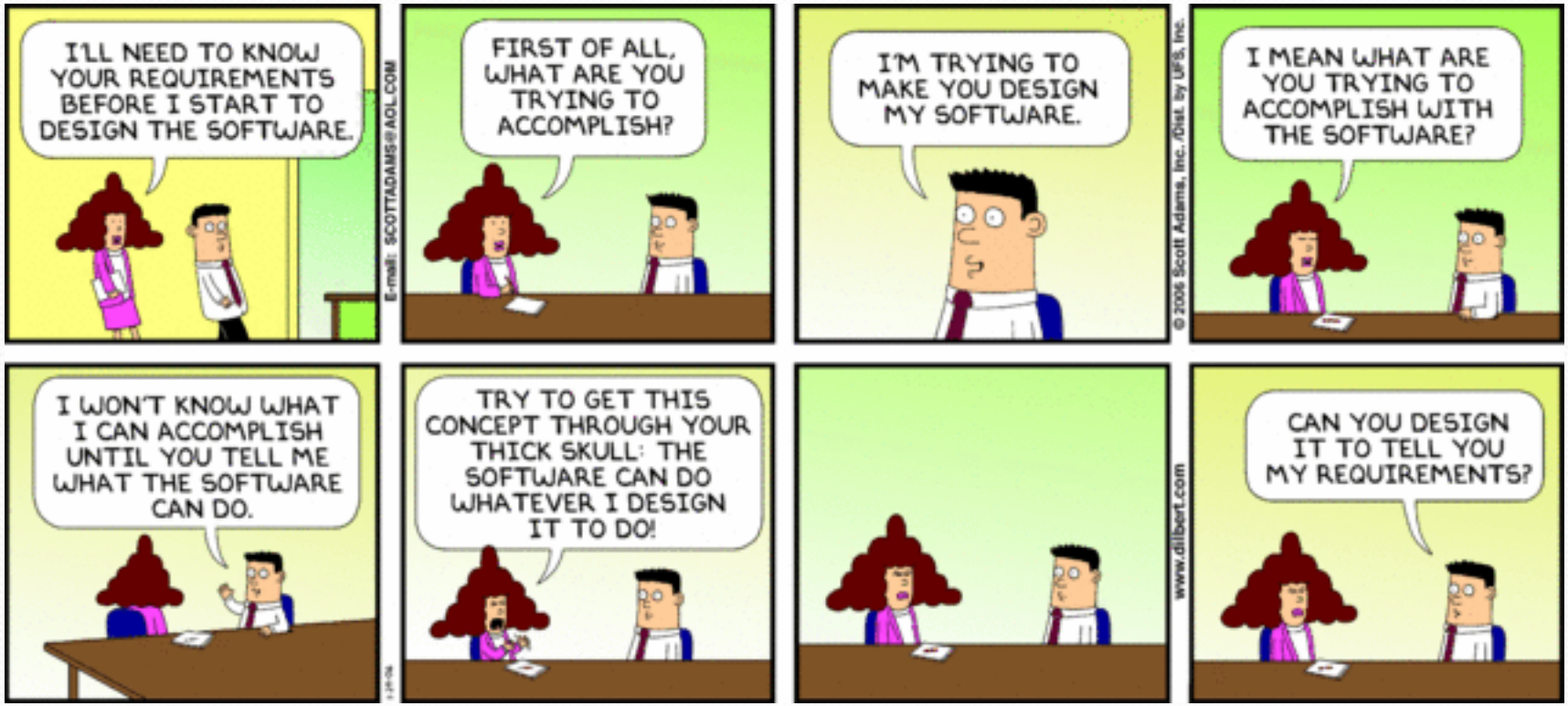
# CS 115, Autumn 2021

Lecture 24: planning; parameters




Thanks to Marty Stepp and Stuart Reges for parts of these slides

# Exercise

- What does the following program output?

```
def main():
    one()
    two()
    three()
    two()

def one():
    print("one first")
    two()
    print("one end")
```

```
def two():
    print("two first")

def three():
    print("three first")
    one()
    two()

main()
```

# Planning

- This program is bigger and more complex than most we have written in class
  - It helps to plan before we start writing code
    - Loop tables
    - Pseudocode
    - Flow charts

# Pseudocode

- Writing out a detailed description of our program in English

```
user guesses a number

if the number is less than 10
  output small number message

otherwise, if the number is between 10 and 80
  output medium number message

otherwise
  output big number message
```

# Pseudocode

- You can start thinking about your pseudocode by asking yourself
  - What do I need?
  - Is there a pattern here I have seen before

- Counting 1s rolled in 10 rolls of dice – List of what we need:
  - `loop – 10 times`
  - `if die value is 1`
  - `cumulative sum`
    - `variable before the loop`
    - `loop`
    - `Setting the variable to itself + something else in the loop`

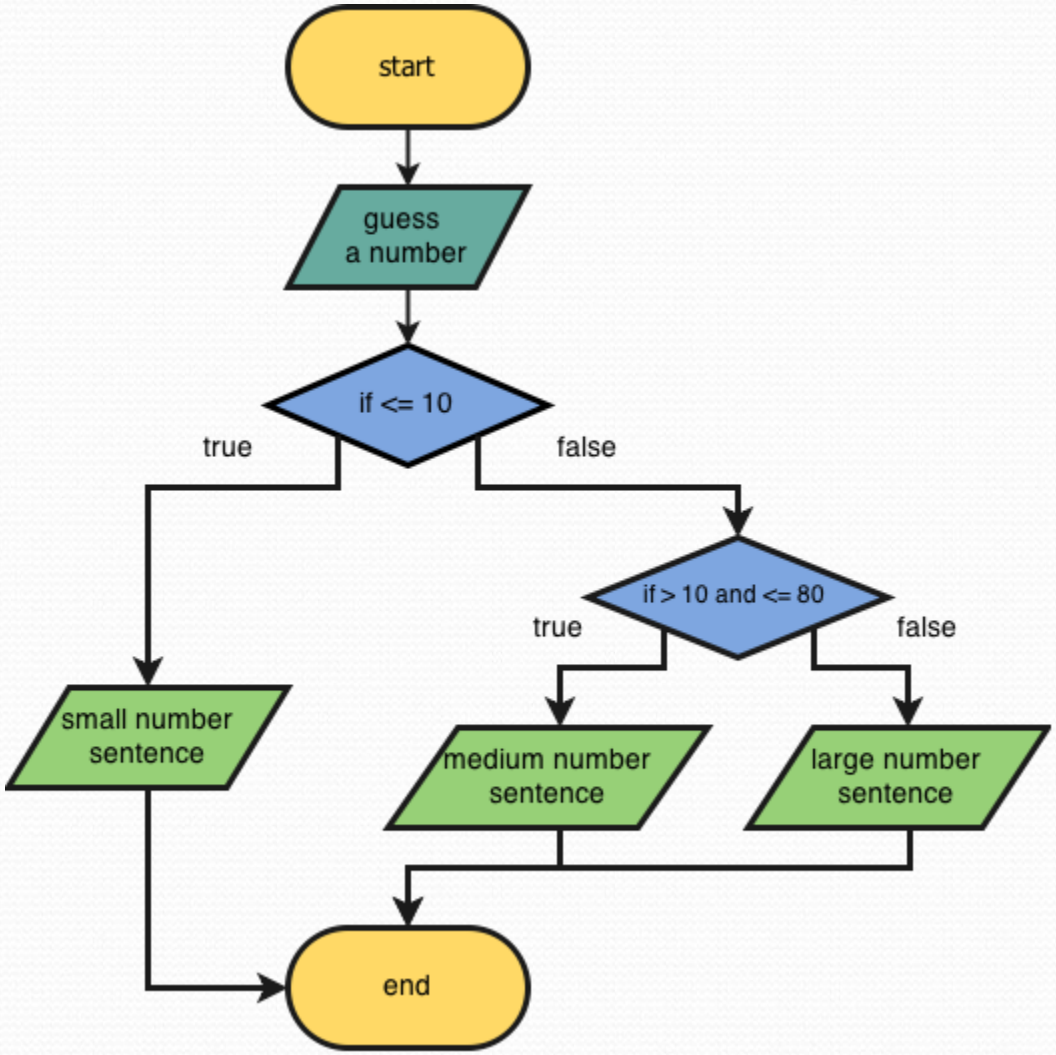
# Pseudocode

```
create counter variable

for 10 times
    roll dice
    if that number is a 1
      set counter variable to counter + 1

print count
```

# Flow charts


start
guess a number
if <= 10
true
false
if > 10 and <= 80
true
false
small number sentence
medium number sentence
large number sentence
end

# Flow chart symbols


- Used for the start and/or end of any block of code


- Used for printing or reading information out of the interface


- Used any time the code might branch based on a decision


- Used for a program instruction that isn't a decision or I/O.


- Used generally for calls to subroutines

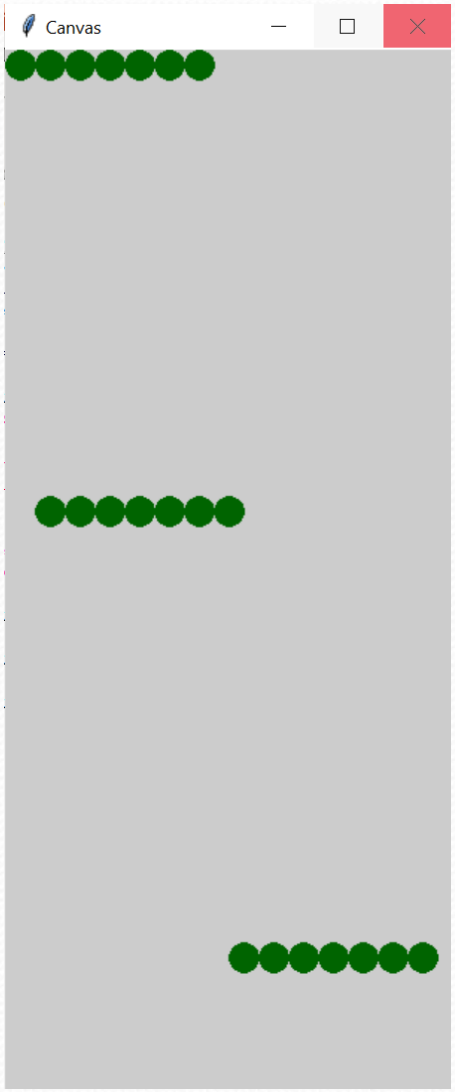
- Used to show an entry to another part of the flowchart

- Used to show the diagram continues on the next page

# What do we need?

# Exercise: `green_circles`

- Alter the code used to produce the picture to the left which contains a separate function for each line of green circles to just contain one green circle drawing function.



# Green circles initial code

```
def main():
    panel = drawing_panel(300, 700, "#CCCCCC")
    green_circles_top(panel)
    green_circles_middle(panel)
    green_circles_bottom(panel)

def green_circles_top(panel):
    for i in range(0, 7):
        panel.fill_oval(i * 20, 0, 20, 20, "green")

def green_circles_middle(panel):
    for i in range(0, 7):
        panel.fill_oval(20 + i * 20, 300, 20, 20, "green")

def green_circles_bottom(panel):
    for i in range(0, 7):
        panel.fill_oval(150 + i * 20, 600, 20, 20, "green")

main()
```

# Green circles solution code

```
def main():
    panel = drawing_panel(300, 700, "#CCCCCC")
    green_circles(panel, 0, 0)
    green_circles(panel, 20, 300)
    green_circles(panel, 150, 600)

def green_circles(panel, x, y):
    for i in range(0 7):
        panel.fill_oval(x + i * 20, y, 20, 20, "green")

main()
```