

ECS10

1/23

- Need an example with two variables changing in a loop. Point out initialization, change, print.

### Example multiple choice

```
mon = 1
while mon < 3:
    print("ha")
```

- This program prints....

### Sentry variable

```
mon = 1
while mon < 3:
    print("ha")
```

- The variable whose value controls when the while loop stops.
- Here mon is the sentry variable.

### Structure of program

- First, get input data.
- Then, a loop that does something with it
- Finally, an ending
  
- Suggestion: ignore the interest at first, just make payments until the balance is gone.
- When you have this working, put in interest.
  
- Lab hours 11-4 tomorrow

### End of program

- Balance coming in \$121.27
- Interest owed \$1.21
- Usual payment \$300
- Instead pay \$122.48
- How do you know you need to make a different payment?
- How do you write this in a Python program?

### Looping for input

- We can keep asking for a reasonable input until we get it.

```
reply = input("Enter rate: ")
while not canBeFloat(reply):
    reply = input("Enter rate: ")
```

### Another version.

- We can keep asking for a reasonable input until we get it.

```
ok = False
while not ok:
    reply = input("Enter rate: ")
    ok = canBeFloat(reply)
```

### Upcoming Events

- Assignment due Thurs night
- Midterm Monday
- Review in class on Friday
- BRING A SCANTRON 2000 FORM
- You can buy them at the Bookstore
- Multiple choice and writing a program
- Sample midterm on Web later today.

### Example multiple choice

```
def main():
    print("a")

    print("b")
    main()
    print("c")
```

- This program prints....

### Looping for multiple choice

- We could have used this in the last program...

### Complicated Boolean expressions

- It's easy to mess these up.
- Possible to make expressions that are never True.

### Example multiple choice

```
answ = "h"  
B = not ( (answ == "h") or (answ == "t") )
```

- After these lines, B will contain....

### This version is easier to get right

```
gotInput = False  
while not gotInput:  
    answ = input("Enter h or t: ")  
    gotInput = (answ == "h") or (answ == "t")  
if not gotInput:  
    print ("Not a valid choice.")
```

### Another way - while-break loop

```
while True:  
    reply = input("Enter h or t:")  
    if (reply == "h") or (reply == "t") :  
        break  
    else:  
        print ("not valid")  
# now reply is either h or t
```

### while-break loop

```
while True:  
    .....  
    if.....:  
        break  
    .....  
□ Loop continues until break statement, then jumps out  
□ return - ends a function  
□ break - ends a loop
```

### Many uses of while loops

- Count a specific number of times

```
while month <= 12:
```

- Wait for valid input

```
while not ok:
```

- Wait for balance to get to zero

```
(your program here)
```