Midterm 1
- Midterm is Friday Oct 14, in class.
  - Bring a Scantron 2000 form
  - 10-12 multiple choice questions – 65 pts
  - Short (8 lines?) program – 35 pts
- Open book, open notes.
- Please contact me beforehand if you have to miss it; only illness counts as a reason.
- Wd's Oct 19 is the drop deadline.

Preparation
- Sample test on SmartSite
- Section – attend one before the test if possible
- Lab Hours - bring questions
- Review lecture slides and programs from class, on course Web page. Read book sections if you have not already.
- Review your programs and the solution programs 2 and 3, on SmartSite.

Data types and operators

- \( x = 10 \)
- \( y = x/3 \)
- \( x \) is an integer, \( y \) is a float

- print "Mary"+"!"]))
- Prints "Mary!" – string concatenation

Input()
- Always returns a string.

```python
feet = Input("Enter feet: ")
inches = feet*12
```
- This is going to crash! You can't multiply a sting by 12.

input()
- Programs often end with
  ```python
  input("Press enter to exit.")
  ```
- This does NOT exit the program. It makes the program wait for the user to type something.
- Why?
### Variables and assignment

- **x = 0**
  - Variable on left.
  - Expression (something which is or computes a data value) on the right:
    - \( x = x / 100.0 \)
  - Uses old value of \( x \) on the right to compute a new data value, which is stored in variable \( x \) on left.

### Boolean expressions

- Either have value True or False (capitalized!)
  - **x == 0**
  - Use it in an if or a while statement.
    - while \( x >= 0 \):
      - \( x = x - \text{payment} \)

### = or == question

- The lines:
  - \( x = 0 \)
  - \( x == 1 \)
  - a) Will result in \( x \) containing the value 0.
  - b) Will result in \( x \) containing the value 1.
  - c) Will result in \( x \) containing the value False.
  - d) Will cause an error.

### Complicated Booleans

- if not ((reply == 'r') or (reply == 'p') or (reply == 's')):
  - True when reply is NOT ‘r’, ‘p’, or ‘s’
- if user == 'r' and user == 'p':
  - Always False, so the block under if never done.
- if reply != 'Y' and reply != 'N':
  - True when reply is not ('R' or 'D')

### randrange function

- It’s in the book and 10/12 lecture. Test it out in IDLE.
  - randrange(0,10)
  - Produces a random number between 0 and 9.
Example if-elif-else question

The following lines:
from random import randrange
points = 0
x = randrange(0,3)
if x == 0:
    print('rock')
elif x <= 1:
    print('paper')
else:
    print('scissors')

a) Will print one of ‘rock’, ‘paper’, ‘scissors’
b) Might print both ‘rock’ and ‘paper’
c) Will never print ‘rock’
d) Might not print anything.

Example if-elif-else question

The following lines:
from random import randrange
points = 0
x = randrange(0,3)
if x == 0:
    print('rock')
elif x <= 1:
    print('paper')
else:
    print('scissors')

a) Will print one of ‘rock’, ‘paper’, ‘scissors’
b) Might print both ‘rock’ and ‘paper’
c) Will never print ‘rock’
d) Might not print anything.

If-elif-else exercise

Write a program that gets a random number between 1 and 4, prints it out, and then correctly identifies it with the sentence “It is four”, “It is three”, “It is two”, or “It is one”, using if-elif-elif-else.

def main() program style

from random import randrange
def main():
    x = randrange(0,4)
    if x == 3:
        return
    print(x)
main()

What does this program do?

Common error

What does this three-line program print?
interest = 3.4
balance = balance + interest
print(balance)

What does this three-line program print?
interest = 3.4
balance = balance + interest
print(balance)

Nothing! It causes an error.
The variable balance is used on the right-hand side before it has something in it.
How many times?

How many lines will this program print?

\[
\begin{align*}
n &= 16 \\
\text{while } n > 1: \\
&\quad n = n/2 \\
&\quad \text{print } n
\end{align*}
\]

Pretend to be the program, and write out what the program will print, on scratch paper.

How many times?

How many lines will this program print?

\[
\begin{align*}
n &= 16 \\
\text{while } n > 1: \\
&\quad n = n/2 \\
&\quad \text{print } n
\end{align*}
\]

Boolean variable

gotAnswer = False
while not gotAnswer:
    answer = input("Enter Y or N: ")
    if answer == "Y" or answer == "N":
        gotAnswer = True

After running this, gotAnswer contains True and answer contains either "Y" or "N".

Programming Problem

Question 1d (Programming Problem – Write your program below and run on the Scintex form)
Write a program that calculates a mass decay by a given percentage every year, and less than 1% of the original mass is left. Here is an example output (use input as underlined):


<table>
<thead>
<tr>
<th>Year</th>
<th>Mass left</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Here is an example output (use input as underlined):

<table>
<thead>
<tr>
<th>Year</th>
<th>Mass left</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Please note comments: in particular, declare input and output.

Approach

- First understand what the program is supposed to do.
- Here, the program is supposed to:
  1. Get user input – initial mass, rate of decay
  2. Reduce mass by (rate of decay) percent
  3. Until mass is reduced to < 1% of its initial mass
- Heart of the program will be a while loop.
- Start in the middle... on scratch paper.

While loop

- The program will require a while loop. You really have to understand how it works.

```python
while Boolean is True:
    do block
```

- while the Boolean test is True, the block under the while statement repeats.
- Somehow the code in the block has to eventually make the Boolean False.
### Step 1: Choose variables
- **initialMass** - The mass the user inputs at the beginning. Float.
- **mass** - The mass that decreases every year. Float.
- **rate** - The percent decay every year. Float.
- **year** - The number of years so far. Integer.

### Step 2: Block under the while
```
while _________:
    mass = mass - mass*rate/100.0
    year = year + 1
    print 'After year',year,
    print 'The mass is', mass
```

- What should the Boolean condition in the while statement be?

### Step 3: The Boolean condition
```
while mass > initialMass*0.01:
    mass = mass - mass*rate/100.0
    year = year + 1
    print 'After year',year,
    print 'The mass is', mass
```

- Make sure that something that is changing in the loop will end up making the condition False eventually (and that it starts off True...)

### Step 4: Getting user input
```
reply = input("Enter initial mass (grams):")
initialMass = float(reply)
reply = input("Enter decay rate in percent: ")
rate = float(reply)
```

- Midterm question does not say that input has to be checked to avoid crashes; but this is required in real life and homework!

### Step 5: Beginning values for other variables
```
mass = initialMass
year = 0
while mass > initialMass/100:
    mass = mass - mass * rate / 100
    year = year+1
    print "After year",year,
    print "the mass is",mass
```

### Step 6: After the loop
```
print "Years needed to drop below 1% of initial mass is",year
```

- Comments! Comments! Comments!
- If you make mistakes in the code, but the comments show what you were trying to do, you might get partial credit.
Prepare for program

☐ Try to re-write this program, following the steps, without looking at the answer.

☐ Do the program on the sample midterm. Try it on paper, and then on a computer; does it work? If you can't do it on the computer, seek help immediately (in section or lab hours).