

ECS10

1/25

### Midterm 1

- Midterm is Monday Jan 28, 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.
- Mon Feb 4 is the **drop deadline**.

### Preparation the weekend

- SmartSite:
  - ▣ Sample test – in Resources
  - ▣ Sample solutions to programs 2 and 3 – in Resources
- Course Web page:
  - ▣ Lecture sildes and program examples
- Textbook:
  - ▣ Chapters 2 and 3
  - ▣ Try the examples
- Review your programs.

### Data types and operators

```
x = 10
y = x/3
```

- What types are these variables?

```
print("Mary"+"!")
```

- What does this print?

### Data types and operators

```
x = 10
y = x/3
```

- x is integer; y is floating point

```
print("Mary"+"!")
```

- Prints "Mary!" – string concatenation

### input()

- What will this program do?

```
inches = input("Enter inches: ")
feet = inches/12
```

## input()

```
inches = input("Enter inches: ")  
feet = inches/12
```

- ❑ Crashes! Inches is a string, you cannot divide it by 12. Even the string "200" cannot be divided by 12.

## input()

- ❑ What will this program do?

```
print("The program is over.")  
input("Press enter to exit")  
print("And now?")
```

## input()

- ❑ What will this program do?

```
print("The program is over.")  
input("Press enter to exit")  
print("And now?")
```

- ❑ It will print all three lines.

## input()

- ❑ Programs often end with `input("Press enter to exit.")`
- ❑ This does NOT exit the program. It makes the program wait for the user to type something.

## 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 - payment
```

- ❑ Can store in a variable.

```
bigEnough = (x > 10)
```

## Boolean value

- What does this print?

```
ready = True
if not ready:
    print("waiting")
else:
    print("going")
```

## = 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 the program to crash.

## = 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 ('Y' or 'N')

## randrange function

- It's in the book and 1/14 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)
```

## Common error

What does this three-line program print?

```
interest = 3.4
balance = balance + interest
print(balance)
```

- Crashes!
- 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?

```
n = 16
while n > 1:
    n = n/2
    print ( n )
```

## How many times?

How many lines will this program print?

```
n = 16
while n > 1:
    n = n/2
    print n
```

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

## Programming Problem

Question 16 (Programming Problem - Write your program below and not on the Scantron form!)

Write a program that simulates a mass decay by a given percentage every year, until less than 1% of the original mass is left. Here is an example output (user input is underlined):

```
Enter initial mass (grams): 100
Enter yearly decay rate in percent (0-100): 50
after year 1 the mass is 50.0 grams
after year 2 the mass is 25.0 grams
after year 3 the mass is 12.5 grams
after year 4 the mass is 6.25 grams
after year 5 the mass is 3.125 grams
after year 6 the mass is 1.5625 grams
after year 7 the mass is 0.78125 grams
Years needed to drop below 1% of initial mass = 7
```

Please use comments, in particular, declare inputs and outputs.

## 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.

**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, otherwise it will be an infinite loop.

## 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 loop 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).