Announcements

• Reminders:
  – MPL Assignment 1 due tonight by 11:59pm
  – Program 1 due tomorrow by 11:55pm

• MemPy User Group
  – Monday, Sept. 15th, 6:30-8pm, in FJ-B.
  – 1 point Extra Credit of Midterm 1 for attending

Code with an If-Statement

```python
# This program calculates your exam average.
exam1 = int(input("What is your first exam score? "))
exam2 = int(input("What is your second exam score? "))
exam3 = int(input("What is your third exam score? "))
average = (exam1 + exam2 + exam3) / 3

choice = input("Did you do the extra assignment? (yes or no)")
if choice == "yes":
    average += 5
print("Your exam average is", average)
```

Saved as exam-if.py in my Public directory

If-Else Example

```python
# This program calculates your exam average.
exam1 = int(input("What is your first exam score? "))
exam2 = int(input("What is your second exam score? "))
exam3 = int(input("What is your third exam score? "))
average = (exam1 + exam2 + exam3) / 3

choice = input("Did you do the extra assignment? ")
if choice == "yes":
    print("Your exam average is", average + 5)
else:
    print("Your exam average is", average)
```

Saved as exam-if-else.py in my Public directory
Comparing Strings

- Strings can be compared using the == and != operators
- String comparisons are case sensitive
- Strings can be compared using >, <, >=, and <=
  - Compared character by character based on the ASCII values for each character
  - If shorter word is substring of longer word, longer word is greater than shorter word

Using String Comparisons

```python
#!/usr/bin/env python
# This program takes in 2 names and prints them out in alphabetical order

name1 = input("Enter name 1: ")
name2 = input("Enter name 2: ")

print("Here are the names, listed alphabetically.")
if name1 < name2:
  print(name1)
  print(name2)
else:
  print(name2)
  print(name1)
```

Saved as compareNames.py in my Public directory

Practice

- Write a program that prompts a user for his or her age and prints out whether or not they are (legally) allowed to drink alcohol.
The if-elif-else Statement

- **if-elif-else statement**: special version of a decision structure
  - Makes logic of nested decision structures simpler to write
  - Can include multiple **elif** statements
- **Syntax**:
  ```python
  if condition1
      statements
  elif condition2
      statements
  else
      statements
  ```

Logical Operators

- **Logical operators**: operators that can be used to create complex Boolean expressions
  - **and** operator and **or** operator: binary operators, connect two Boolean expressions into a compound Boolean expression
  - **not** operator: unary operator, reverses the truth of its Boolean operand
The and Operator

if ______ and ________:
    # do something
else:
    # do something else

Truth table for the and operator

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value of the Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>false and false</td>
<td>false</td>
</tr>
<tr>
<td>false and true</td>
<td>false</td>
</tr>
<tr>
<td>true and false</td>
<td>false</td>
</tr>
<tr>
<td>true and true</td>
<td>true</td>
</tr>
</tbody>
</table>

Both individual tests must be True to make the entire if statement True.

The or Operator

if ______ or ________:
    # do something
else:
    # do something else

Truth table for the or operator

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value of the Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>false and false</td>
<td>false</td>
</tr>
<tr>
<td>false and true</td>
<td>true</td>
</tr>
<tr>
<td>true and false</td>
<td>true</td>
</tr>
<tr>
<td>true and true</td>
<td>true</td>
</tr>
</tbody>
</table>

Either (or both) individual tests must be True to make the entire if statement True.

Short-Circuit Evaluation

- Short circuit evaluation: deciding the value of a compound Boolean expression after evaluating only one sub expression
  - Performed by the or and and operators
    - For or operator: If left operand is true, compound expression is true. Otherwise, evaluate right operand
    - For and operator: If left operand is false, compound expression is false. Otherwise, evaluate right operand

The not Operator

- Takes a Boolean expression as operand and reverses its logical value
  - Sometimes it may be necessary to place parentheses around an expression to clarify to what you are applying the not operator

Truth table for the not operator

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value of the Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
Checking Numeric Ranges with Logical Operators

- To determine whether a numeric value is within a specific range of values, use `and`
  - Example: \( x \geq 10 \) and \( x \leq 20 \)
- To determine whether a numeric value is outside of a specific range of values, use `or`
  - Example: \( x < 10 \) or \( x > 20 \)

Boolean Variables

- **Boolean variable**: references one of two values, True or False
  - Represented by `bool` data type
- **Commonly used as flags**
  - Flag: variable that signals when some condition exists in a program
    - Flag set to False \( \rightarrow \) condition does not exist
    - Flag set to True \( \rightarrow \) condition exists

```python
# This program determines whether a bank customer qualifies for a loan.
# Constants for minimum salary and minimum number of years on the job
MIN_SALARY = 30000.0
MIN_YEARS = 2

# Get the customer's annual salary.
salary = float(input('Enter your annual salary: '))

# Get the number of years on the current job.
years_on_job = int(input('Enter the number of years you have been employed: '))

# Determine whether the customer qualifies.
if salary >= MIN_SALARY:
    if years_on_job >= MIN_YEARS:
        print('You qualify for the loan.
    else:
        print('You must have been employed for at least 2 years to qualify.
else:
    print('You must earn at least $30,000 per year to qualify.
```
# This program determines whether a bank customer qualifies for a loan.
# Constants for minimum salary and minimum
# years on the job.
MIN_SALARY = 80000.0
MIN_YEARS = 2

# Get the customer's annual salary.
salary = float(input('Enter your annual salary: '))

# Get the number of years on the current job.
years_on_job = int(input('Enter the number of years employed: '))

# Determine whether the customer qualifies.
if salary >= MIN_SALARY and years_on_job >= MIN_YEARS:
    print('You qualify for the loan.')
else:
    print('You do not qualify for this loan.')

## Review Questions

1. Does an if statement always need to be followed by an else statement?

2. If you write an if-else statement, under what circumstances do the statements that appear after the else clause execute?

3. Assume the variables a = 2, b = 4, c = 6. What do the following statements evaluate to (true or false)?
   a) a == 4 or b > 2
   b) 6 <= c and a > 3
   c) 1 != b and c != 3
   d) a >= -1 or a <= b
   e) not (a > 2)

## Practice

Write a program that asks the user when their birthday is (month and day of month as ints). Then print a message telling them if their birthday has already passed this year, is yet to come this year, or is today.

## Next Time

- Functions
- Section 5.1-5.3