CSCE 110
Programming I

David Kebo Hougninou

Basics of Python

Decision-making statements
Conditional Execution and Flags

Instructions can be made to execute conditionally

<table>
<thead>
<tr>
<th>CMP</th>
<th>r3,#0</th>
<th>CMP</th>
<th>r3,#0</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEQ</td>
<td>skip</td>
<td>ADDNE</td>
<td>r0,r1,r2</td>
</tr>
<tr>
<td>ADD</td>
<td>r0,r1,r2</td>
<td>SKIP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ADDNE r0,r1,r2</td>
</tr>
</tbody>
</table>

Conditional flowchart

TRUE

FALSE

Conditional / Selection
## Conditional Codes Examples

### C source code

```c
if (r0 == 0)
{
    r1 = r1 + 1;
}
else
{
    r2 = r2 + 1;
}
```

### Unconditional

- `CMP r0, #0`
- `BNE else`
- `ADD r1, r1, #1`
- `B end`
- `ADD r2, r2, #1`
- `end ...

### Conditional

- `CMP r0, #0`
- `ADDEQ r1, r1, #1`
- `ADDNE r2, r2, #1`
- `...`

- 5 instructions
- 3 instructions

## Conditional Execution

![Flowchart](image)

- `if <condition>`
- `<do this>`

## Conditional / Selection
Conditional Execution

```python
if <condition>:
    <do this>
```

Indentation is mandatory
One tab

Conditional / Selection

**Flowchart:**
- **TRUE**:
  - `<condition>`
  - `<do this>`
- **FALSE**:
  - `<condition>`
  - `<do that>`

```python
if <condition>:
    <do this>
else:
    <do that>
```

Conditional / Selection
Activity

```
1 x = int(input('Enter the value of x: '))
2 if (x > 100):
3     print('A')
4 else:
5     print('B')
6 print('C')
```

What is the expected output for the following values of x?

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = 150</td>
<td>A C</td>
</tr>
<tr>
<td>x = 100</td>
<td>B C</td>
</tr>
<tr>
<td>x = 98</td>
<td>B C</td>
</tr>
<tr>
<td>x = 0</td>
<td>B C</td>
</tr>
</tbody>
</table>

Activity solution

```
1 x = int(input('Enter the value of x: '))
2 if (x > 100):
3     print('A')
4 else:
5     print('B')
6 print('C')
```

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = 150</td>
<td>A C</td>
</tr>
<tr>
<td>x = 100</td>
<td>B C</td>
</tr>
<tr>
<td>x = 98</td>
<td>B C</td>
</tr>
<tr>
<td>x = 0</td>
<td>B C</td>
</tr>
</tbody>
</table>
What is the expected output for the following values of x?

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = 150</td>
<td>A</td>
</tr>
<tr>
<td>x = 100</td>
<td>B  C</td>
</tr>
<tr>
<td>x = 98</td>
<td>B  C</td>
</tr>
<tr>
<td>x = 0</td>
<td>B  C</td>
</tr>
</tbody>
</table>
# Truth Value Testing

<table>
<thead>
<tr>
<th>False values</th>
<th>Truth values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>False</td>
<td></td>
</tr>
<tr>
<td>'' - an empty string</td>
<td></td>
</tr>
<tr>
<td>[] - an empty list</td>
<td></td>
</tr>
<tr>
<td>{} - an empty dictionary</td>
<td></td>
</tr>
<tr>
<td>() - an empty tuple</td>
<td></td>
</tr>
<tr>
<td>set() - an empty set</td>
<td></td>
</tr>
<tr>
<td>range(0) - an empty range</td>
<td></td>
</tr>
</tbody>
</table>

Everything else

# Comparison Operators

<table>
<thead>
<tr>
<th>Description</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strictly less than</td>
<td>&lt;</td>
</tr>
<tr>
<td>Less than or equal to</td>
<td>&lt;=</td>
</tr>
<tr>
<td>Strictly greater than</td>
<td>&gt;</td>
</tr>
<tr>
<td>Greater than or equal</td>
<td>&gt;=</td>
</tr>
<tr>
<td>Equal to</td>
<td>==</td>
</tr>
<tr>
<td>Not equal to</td>
<td>!=</td>
</tr>
</tbody>
</table>
Comparison Operators

Comparison Operators

Value 1 \(\Rightarrow\) Comparison Operator \(\Rightarrow\) Value 2

\[= \neq > < \\\]

Value 1

Comparison Operator

Value 2

True or False

Comparison Operators

Python shell examples.

\[
>>> 9 < 100 \\
True \\
>>> 10 != 15 \\
True \\
>>> 10 == 15 \\
False
\]
### Activity 2.1

What is the expected output for the following values of x?

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{2} &lt; 3 \times 2$</td>
<td>True</td>
</tr>
<tr>
<td>$9^{0.5} \neq 3$</td>
<td>False</td>
</tr>
<tr>
<td>size = 10</td>
<td>False</td>
</tr>
<tr>
<td>$2 \times $size &lt; $size$</td>
<td>False</td>
</tr>
<tr>
<td>$1089 % 10 = 89$</td>
<td>False</td>
</tr>
<tr>
<td>len(&quot;eight&quot;) == 8</td>
<td>False</td>
</tr>
<tr>
<td>bool(&quot;&quot;) == False</td>
<td>True</td>
</tr>
</tbody>
</table>
Logical Operators

<table>
<thead>
<tr>
<th>Description</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical AND</td>
<td>and</td>
</tr>
<tr>
<td>Logical OR</td>
<td>or</td>
</tr>
<tr>
<td>Logical NOT</td>
<td>not</td>
</tr>
</tbody>
</table>

Logical Operators

Value 1 → Logical Operator and or not → Value 2

True or False
Logical operators

and, or, not

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A and B</th>
<th>A</th>
<th>B</th>
<th>A or B</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>not A</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>
Membership Operators

<table>
<thead>
<tr>
<th>Description</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>True if a variable is in the sequence; false otherwise.</td>
<td>in</td>
</tr>
<tr>
<td>True if a variable is not in the specified sequence; false otherwise.</td>
<td>not in</td>
</tr>
</tbody>
</table>

Activity 2.2

Write a program that takes an integer and:
If the integer is a multiple of 3, print "Howdy".
If the integer is a multiple of 5, print "Aggies".
If the integer is a multiple of both 3 and 5, print "Howdy Aggies".
Activity 2.3

Create a program that takes two strings as arguments and return either True or False depending on whether the total number of characters in the first string is equal to the total number of characters in the second string.

Activity 2.4

Create a program that takes a number as an argument and prints True or False depending on whether the number is symmetrical or not.

A number is symmetrical when it is the same as its reverse.
Activity 2.5

Create a program that takes three integer values \((a, b, c)\) and prints the number of equal values.

Operators precedence

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>Exponentiation (raise to the power)</td>
</tr>
<tr>
<td>-</td>
<td>minus</td>
</tr>
<tr>
<td>* / % //</td>
<td>Multiplication, division, remainder and floor division</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition and subtraction</td>
</tr>
<tr>
<td>&lt;,&lt;=,&gt;,&gt;=,==, !=</td>
<td>Comparison</td>
</tr>
<tr>
<td>not</td>
<td>not logical</td>
</tr>
<tr>
<td>and</td>
<td>and logical</td>
</tr>
<tr>
<td>or</td>
<td>or logical</td>
</tr>
</tbody>
</table>
Example

Write a Python program to implement one of Einstein’s favorite puzzles:

1. Enter a 3 digit number. The first and last digit differ by at least two.
   e.g. 442 is correct but 244 is not since the first and last digit differ by -2.
2. Reverse the input.
3. Subtract the reversed number from the original number.
4. Reverse the difference.
5. Add the difference to the reversed difference.
6. The sum should be 1089.

Example

Your program should test if the user entered a three digits number.
If the number doesn’t have three digits the program should display the message ‘The input is not valid’.
Example: solution

```python
number = input('Enter a number of 3 digits: ')
number_reverse = number[::-1]
result = int(number) - int(number_reverse)
result_reverse = str(result)[::-1]
total = int(result) + int(result_reverse)
print('The output is ', total)
```

Example: solution

```python
number = input('Enter a number of 3 digits: ')
if (len(number) == 3):
    number_reverse = number[::-1]
    result = int(number) - int(number_reverse)
    result_reverse = str(result)[::-1]
    total = int(result) + int(result_reverse)
    print('The output is ', total)
else:
    print('The input is not valid. You should enter a 3 digit number.')```
Example: solution

```python
number = input('Enter a number of 3 digits: ')

if (len(number) == 3):
    number_reverse = number[::-1]
    result = int(number) - int(number_reverse)
    result_reverse = str(result)[::-1]
    total = int(result) + int(result_reverse)
    print('The output is ', total)
else:
    print('The input is not valid. You should enter a 3 digit number. ')
```

Example: solution

```python
number = int(input('Enter a number of 3 digits: '))

# Testing the input
if ((number>=100) and (number<=999)):
    # Reversing the number
    h = number // 100
    t = (number % 100)//10
    o = number % 10
    number_reverse = o*100 + t*10 + h
    result = number - number_reverse
    # Reversing the number
    h = result // 100
    t = (result % 100)//10
    o = result % 10
    result_reverse = o*100 + t*10 + h
    total = result + result_reverse
    print('The output is ', total)
else:
    print('The input is not valid. You should enter a 3 digit number. ')
```
Example

Consider a second degree polynomial or Quadratic equation in the form:

\[ ax^2 + bx + c = 0 \]

Write a program to test if a quadratic equation has real roots.

Conditional Execution

```
if <condition 1>:
    <do this 1>
else:
    if <condition 2>:
        <do this 2>
    else:
        <do that>
```
How to test if a quadratic equation has real roots?

Example: discriminant

\[ x^2 + 3x + 2 = 0 \]

\[ a = 1, \ b = 3, \ c = 2 \]

\[ \Delta = b^2 - 4ac = 3^2 - (4 \times 1 \times 2) = 1 \]
Example

Consider a second degree polynomial or Quadratic equation in the form:

\[ ax^2 + bx + c = 0 \]

Write a program to compute the number of real roots of a quadratic equation.

```python
a = int(input('Enter a: '))
b = int(input('Enter b: '))
c = int(input('Enter c: '))
print('Real roots ?')
if (b**2 - 4*a*c >= 0):
    print('Yes')
else:
    print('No')
```
Example

How can I test the number of real roots of a quadratic equation?

\[ ax^2 + bx + c = 0 \]

is \( b^2 - 4ac \geq 0 \)

```python
1 a = int(input('Enter a: '))
2 b = int(input('Enter b: '))
3 c = int(input('Enter c: '))
4 d = b**2 - 4*a*c
5 if (d > 0):
   6   print('2 real roots')
   7 else:
   8     if (d == 0):
   9        print('1 real root')
 10     else:
 11       print('None')
```
Example

For equation: $x^2 + 3x + 2 = 0$

What is the execution order of this program?

What is the output of this program?

Execution order: 1, 2, 3, 4, 5, 6

Output: 2 real roots

```
1 a = int(input('Enter a: '))
2 b = int(input('Enter b: '))
3 c = int(input('Enter c: '))
4 d = b**2 - 4*a*c
5 if (d > 0):
6     print('2 real roots')
7 else:
8     if (d == 0):
9         print('1 real root')
10    else:
11       print('None')
```
Activity 4.1 solution

For equation: \( x^2 + 2x + 1 = 0 \)

What is the execution order of this program?

What is the output of this program?

Execution order: 1, 2, 3, 4, 5, 7, 8, 9

Output: 1 real root

Activity 4.2

For equation: \( x^2 + 2x + 2 = 0 \)

What is the execution order of this program?

What is the output of this program?

Execution order:

Output:
Activity 4.2 solution

For equation: \( x^2 + 2x + 2 = 0 \)

What is the execution order of this program?

What is the output of this program?

Execution order: 1, 2, 3, 4, 5, 7, 8, 10, 11

Output: None

Conditional Execution

If you need more than two conditions (if-else), then use the \texttt{elif} statement.

```
if <condition>:
    <do this>
elif <condition2>:
    <do this2>
elif <condition3>:
    <do this3>
else:
    <do that>
```
Conditional Execution

Same algorithm - different implementation

```
1 a = int(input('Enter a: '))
2 b = int(input('Enter b: '))
3 c = int(input('Enter c: '))
4 d = b**2 - 4*a*c
5 if (d > 0):
6     print('2 real roots')
7 else:
8     if (d == 0):
9         print('1 real root')
10    else:
11     print('None')
```

Example

Write a program that computes the letter grade of a student given her score grade.

<table>
<thead>
<tr>
<th>Score Grade</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90-100</td>
<td>'A'</td>
</tr>
<tr>
<td>80-89</td>
<td>'B'</td>
</tr>
<tr>
<td>70-79</td>
<td>'C'</td>
</tr>
<tr>
<td>60-69</td>
<td>'D'</td>
</tr>
<tr>
<td>&lt;= 59</td>
<td>'F'</td>
</tr>
</tbody>
</table>
How many conditions does the program test?

5
Conditional Execution

```python
score = int(input('Enter your score: '))
if score >= 90:
    letter = 'A'
else:
    if score >= 80:
        letter = 'B'
    else:
        if score >= 70:
            letter = 'C'
        else:
            if score >= 60:
                letter = 'D'
            else:
                letter = 'F'
print ('Letter grade:', letter)
```

How many conditions does the program test?
It depends on the score!

What is the execution order for a score of 94?

What is the execution order for a score of 83?
Conditional Execution

If you need more than two conditions (if-else), then use the if-elif-else statement.

With this statement, you can have as many elif statement as needed.

```python
if <condition>:
    <do this>
elif <condition2>:
    <do this2>
elif <condition3>:
    <do this3>
else:
    <do that>
```

What is the execution order for a score of 64?

```python
score = int(input('Enter your score: '))

if score >= 90:
    letter = 'A'
elif score >= 80:
    letter = 'B'
elif score >= 70:
    letter = 'C'
elif score >= 60:
    letter = 'D'
else:
    letter = 'F'
print ('Letter grade:', letter)
```
Activity 4.3

```python
score = int(input('Enter your score: '))
if score >= 90:
    letter = 'A'
elif score >= 80:
    letter = 'B'
elif score >= 70:
    letter = 'C'
elif score >= 60:
    letter = 'D'
else:
    letter = 'F'
print ('Letter grade:', letter)
```

a. What is the execution order for a score of 94?

b. What is the execution order for a score of 70?

c. What is the execution order for a score of 0?

Activity 4.3 solution

```python
score = int(input('Enter your score: '))
if score >= 90:
    letter = 'A'
elif score >= 80:
    letter = 'B'
elif score >= 70:
    letter = 'C'
elif score >= 60:
    letter = 'D'
else:
    letter = 'F'
print ('Letter grade:', letter)
```

What is the execution order for a score of 94?

Execution order: 1, 3, 4, 14

Output: A
Example

Write a program that asks for a year and classify the year as **leap year** or **not leap year**.

Leap years are divisible by 4.

However, years divisible by 100 are not leap years unless they are also divisible by 400.
Example

```python
year = int(input('Enter the year: '))
if (year % 4 == 0):
    if (year % 100 == 0):
        if (year % 400 == 0):
            print('Leap Year')
        else:
            print('Not Leap Year')
    else:
        print('Leap Year')
else:
    print('Not Leap Year')
```
Example solution

```python
year = int(input('Enter the year: '))

if (year % 4 != 0):
    print('Not a leap year')
else:
    if (year % 100 != 0):
        print('Leap Year')
    else:
        if (year % 400 != 0):
            print('Not Leap Year')
        else:
            print('Leap Year')
```

Example solution

```python
year = int(input('Enter the year: '))

if (year % 4 != 0):
    print('Not a leap year')
elif (year % 100 != 0):
    print('Leap Year')
elif (year % 400 != 0):
    print('Not Leap Year')
else:
    print('Leap Year')
```
Preferred solution

```python
year = int(input('Enter the year: '))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print('Leap year')
else:
    print('Not Leap Year')
```

References

1. Flowchart In Programming:

2. Explanation of the Einstein's Puzzle:
   http://www.pleacher.com/mp/puzzles/tricks/p1089.html

3. Python Basic Operators:
   https://www.tutorialspoint.com/python/python_basic_operators.htm