CPSC203: WEEK-2 LAB-1
PROBLEM SOLVING

-Prepared By
Nashad Ahmed Safa
Graduate Student
Department of Computer Science

COURSE WEBSITE

http://wiki.ucalgary.ca/page/Courses/Computer_Science/CPSC_203/CPSC_203_Template

INDENTATION IN JES

- Jython uses **whitespace indentation**, rather than curly braces or keywords, to delimit blocks (a feature also known as the off-side rule). An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.
- Indentation is needed when defining-
  - Functions
  - Loops
  - Conditions etc

FUNCTIONS

- What are functions? Well, in effect, functions are little self-contained programs that perform a specific task, which you can incorporate into your own, larger programs. After you have created a function, you can use it at any time, in any place. This saves you the time and effort of having to retell the computer what to do every time it does a common task, for example getting the user to type something in. We will talk about functions in detail in the next class.

LOOPS

- We will look at an example of "looping", which is a technique to enumerate through a list of data repeatedly.
- The **for statement** begins with a header line that specifies an assignment target or targets, along with an object you want to step through. The header is followed by a block of indented statements which you want to repeat. The general format of a for statement looks like the following:

```
for target in object:
    statements
```

LOOPS

Example-1:

```
list = [2, 4, 6, 8]
sum = 0
for num in list:
        sum = sum + num
print "The sum is:", sum
```

Example-2:

```
for count in range(1, 11):
    print count
```

LOOPS

Example-3: Loops through the list having the numbers 1, 3, 6, and 9, and calculates and prints their product.

```
def looping():
    product = 1
    for x in [1, 3, 6, 9]:
        product = product * x
        print "Current product = ", product
    print "Final product = ", product
```

LOOPS

Example-4: Loops through the numbers from 4 to 1, while subtracting 1 in every iteration

```
def looping():
    for i in range(4,1,-1):
        print i
```

Example-5: Loops through the numbers from 6 to 16 and print the result of dividing each number by 2

```
def looping2():
  for x in range(6,16):
        print x/2.0
```

WHILE LOOP

- We will look at another example of "looping", through the while loop. The general format of a while loop looks like the following:

```
while Boolean-condition:
    body-of-while
```

(body-of-while is repeated as long as Boolean condition is true.)

WHILE LOOP

Example-1: Displays the number starting from 5 in descending order, while the displayed number is greater than zero

```
def looping():
    i = 5
   while i > 0:
          print i
          i -= 1
```

CONDITIONS

- Similar to the concepts learned in Excel and Access, you will learn to implement an if condition.
- The general format of an if looks like the following:

```
if test1
        statements1
elif test2
        statements2
else
        statements3
```

CONDITIONS

- Different operators can be used in the condition of an if statement:

== equal

!= not equal

< less than

<= less than or equal

> greater than

>= greater than or equal

and

or

not

CONDITIONS

- Example 1: In the code below a test to determine the response to what the conditions of the weather is outside.

```
if weather == 'sunny':
        print "Nice weather”
elif weather == 'raining':
        print "Bad weather”
else:
        print "Uncertain, don't plan anything"
```

CONDITIONS

- Example 2: The code below checks whether x is less than or greater than zero

```
def Try_if(x):
        if x< 0:
                print x, "is less than zero"
        elif x == 0:
                print x, "is equal to zero"
        else:
                print x, "is not less than zero"
```

GETTING USER INPUT

For getting user input we can use the following built in function of JES:

variable=**raw_input**(“Prompt”)

Example:

```
name = raw_input("Hi.  What's your name? ")print nam
e
print "Hi, " + name
```