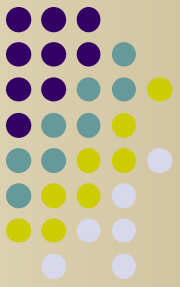


ENGR/CS 101 CS Session

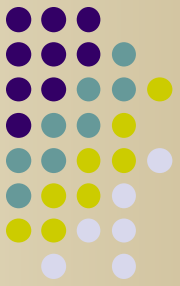
Lecture 3

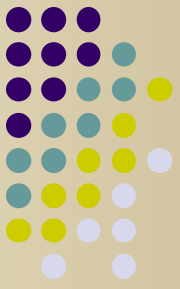


- Log into Windows/ACENET (reboot if in lab machine is in Linux)
- Start IDLE (Python GUI) 2.7
- Open the program file from last class.
- Save this file with a new name like 'lecture03.py'

Outline

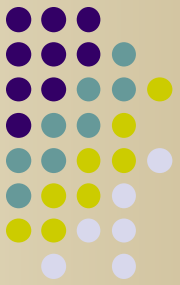
- Functions
- Input loops



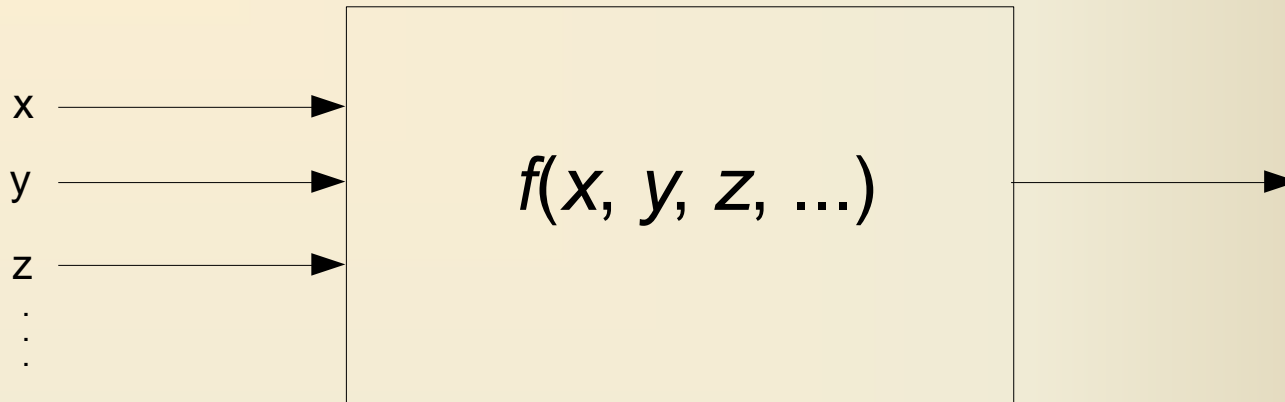


Abstraction

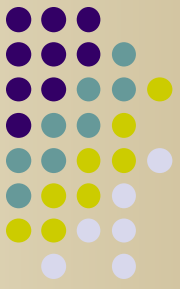
- ***Abstraction*** is defined as a concept or idea not associated with any specific instance. Abstractions are used in many ways in computer science.
- Functions on a calculator are abstractions. They encapsulate computational ideas such as square root, sine, cosine, etc., that work on ***any*** number. But we do not know how the result is computed.



Functions



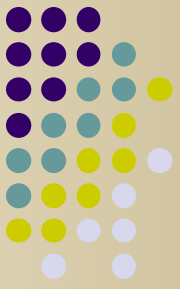
- Consider a (mathematical) function as a black box that **receives** data and **returns** an unnamed result.
- Programming languages can define functions, too.



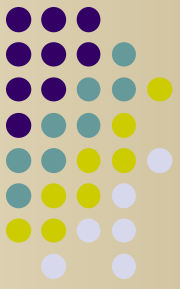
Function Specification

- A function is like a mini-program. We ask the same questions as when designing the main program.
- Example: Write a function that computes and returns the volume of a right cylinder given its radius and its height.

Function Analysis & Design

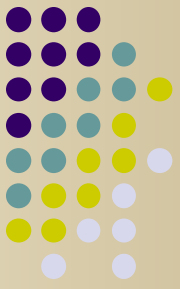


- Analysis: what data is needed and how does it move through the function?
 - Received data comes into the function: radius, height
 - Returned result leaves the function: volume
 - Local data stays within the function: π , base area



Function Analysis & Design

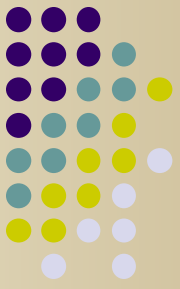
- Design: what algorithm is used to compute result?
 1. Compute area of base of the cylinder
 2. Compute the volume of the cylinder
 3. Return the volume of the cylinder



Python Functions

- Python function declaration syntax is:

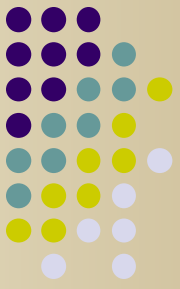
```
def <name>(<parameter list>):  
    <computation statements>  
    return <result>    // if needed
```
- **Parameters** are the names given to the received data. The list consists of variable names separated by commas.
- Body of function must be ***indented***. Python uses indenting to group statements. Use Tab key.



Python Functions

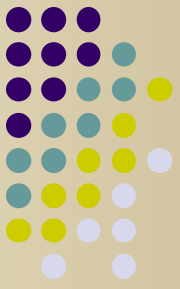
- Type the code for the example function at the top of your program file.

```
# function to compute volume of cylinder
def computeCylinderVolume (baseRadius, cylinderHeight):
    # mathematical constant
    pi = 3.14159
    # 1. Compute base area of cylinder
    baseArea = pi * baseRadius ** 2
    # 2. Compute cylinder volume
    cylinderVolume = baseArea * cylinderHeight
    # 3. Return volume
    return cylinderVolume
```



Function Call

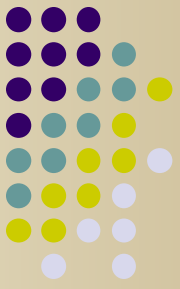
- As seen with the input function, a function is used by **calling** it with **arguments** that are the data being received by the function.
- The arguments, if any, correspond to the parameters by position. The argument values are used to initialize the parameter variables before the function is executed.
- If a function returns a result, it must be saved using an assignment statement.



Main Function

- In most programming languages, the main program also must be in a function.
- We can do this in Python by defining a function called **main**.

```
# main program function
def main():
    # indent main program statements.
    radius = input ('Enter the radius')
    height = input ('Enter the height')
    ...
```

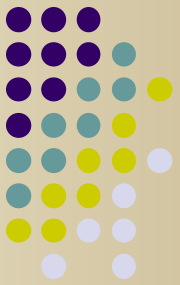


Full Python Program

- Modify the class program to use the function to compute the volume of the cylinder and encapsulate the main program into a main function as shown on the next slide.
- Save the program file and run the module. At the shell prompt type:

```
>>> main()
```

to call the main function and run the program.



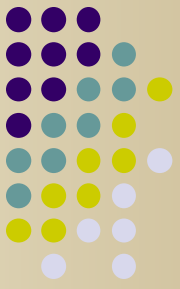
Full Python Program

```
# function to compute volume of cylinder
# ... given on a previous slide

# main program function
def main():
    # 1. Ask user for radius and height
    radius = input('Enter the radius: ')
    height = input('Enter the height: ')

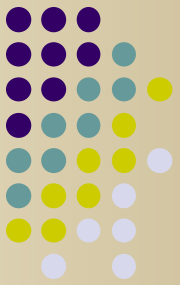
    # 2. Compute volume by calling the function
    volume = computeCylinderVolume(radius,height)

    # 3. Display the cylinder volume
    print 'Cylinder volume is', volume
```



Input Loops

- Sometimes we would like to repeat a computation. To do this we put the computation in a ***loop***.
- One type of loop is an input loop that will repeat until the user types in a special value.
- We can use this loop to test our function without having to start the program each time.



Input Loops

- Example: compute volumes until 0 entered for radius of base

```
Enter the radius of the base (0 to quit): 8
```

```
Enter the height of the cylinder: 16
```

```
Cylinder volume is 3216.98816
```

```
Enter the radius of the base (0 to quit): 6
```

```
Enter the height of the cylinder: 12
```

```
Cylinder volume is 1357.16688
```

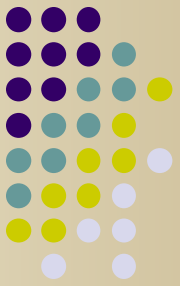
```
Enter the radius of the base (0 to quit): 12
```

```
Enter the height of the cylinder: 6
```

```
Cylinder volume is 2714.33376
```

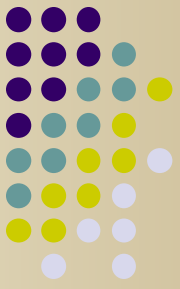
```
Enter the radius of the base (0 to quit): 0
```

```
All done
```



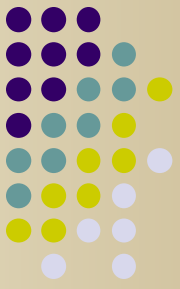
Input Loop Design

- An input loop uses a ***while-loop*** construct. The generic design would be:
 1. Ask the user for an input
 2. While the input is not the special value
 - a. Do the steps that need to be repeated
 - b. Ask the user for another input



Example Design

- For our example program, the design would be:
 1. Ask the user for a radius
 2. While the radius is not 0
 - a. Ask the user for a height
 - b. Compute the volume using the function
 - c. Display the volume
 - d. Ask the user for another radius

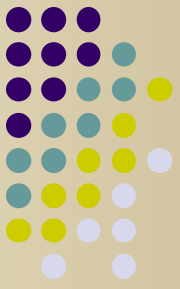


Python Input Loops

- The syntax for a Python input loop is

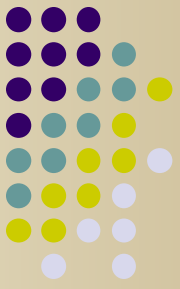
```
<var> = input(<prompt>)  
while <loop test using var>:  
    <steps to be repeated>  
    <var> = input(<prompt>)
```

- Body of loop must be ***indented***. (Again, use the TAB key.)



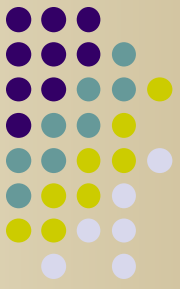
Loop Tests

- Loop tests also are called ***conditions***. They are expressions that evaluate to true or false.
- When a while-loop condition evaluates to true the body is executed. When the condition evaluates to false, the loop ends and the statement after the loop body is executed.
- More on conditions later. The syntax for testing for a non-zero value is: `<var> != 0`



Revised Python Program

- Modify the example program main function to use an input loop that stops when the user enters 0 for the radius as shown on the next slide.
- Save the program and run it.



Revised Python Main Function

```
def main():
    # 1. Ask user for radius
    radius = input('Enter the radius (0 to quit): ')
    # 2. Loop while radius is not 0
    while radius != 0:
        # a. Ask user for height
        height = input('Enter the height: ')
        # b. Compute volume by calling function
        volume = computeCylinderVolume(radius,height)
        # c. Display the cylinder volume
        print 'Cylinder volume is', volume
        # d. Ask user for radius
        radius = input('Enter the radius (0 to quit): ')
    # 3. Statement executed after loop
    print 'All done'
```