A First Program

CSE 1310 – Introduction to Computers and Programming
Alexandra Stefan
University of Texas at Arlington

Credits: a significant part of this material has been created by Dr. Darin Brezeale and Dr. Gian Luca Mariottini

Last updated: 1/15/14
Outline

• Run Python (start the Python IDLE)
• Get familiar with the interpreter (IDLE/Shell)
• Developing code
  – Write a first program
  – Run it
  – Test it
  – Debug it
• How to submit your homework
Simplest Code: Numerical Calculations

• Start the Python shell. You see a welcoming message and the command prompt.

Python 3.2.3 (default, Apr 11 2012, 07:12:16) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.

>>> 

Terminology: we will call >>> “the command prompt”. This is Python’s way of telling you “I am waiting for your input”.
Simplest Code: Numerical Calculations

• Let’s type in a single number, and press ENTER.

Python 3.2.3 (default, Apr 11 2012, 07:12:16) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.

>>> 14
Simplest Code: Numerical Calculations

• After we press ENTER, the computer evaluates what we just typed, and prints the result.

Python 3.2.3 (default, Apr 11 2012, 07:12:16) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.

>>> 14
14
>>>
Simplest Code: Numerical Calculations

• After we press ENTER, the computer evaluates what we just typed, and prints the result.

```python
>>> 14
14
>>> 14
>>> 14
```

• This is not very exciting yet, the computer did not tell us anything we did not know.
Circumference and Area of Circle

• Computing the circumference of a circle with radius = 20.231234:
  – Circumference = radius * pi * 2
  >>> 20.231234 * 3.14159 * 2
  127.11648484412

• Computing the area of the same circle:
  – area = (radius ** 2) * pi
  >>> (20.231234 ** 2) * 3.14159
  1285.8616750694227
Improving the code

```python
>>> 20.231234 * 3.14159 * 2
>>> (20.231234 ** 2) * 3.14159
```

- Tedious to type in long numbers repeatedly.
- The above lines are hard to “read” (here ‘read’ means “infer what the lines of code are trying to achieve”).
- Can you change these lines, to address the problems above?
  - Hint: think of using variables
Using Variables

• Replace:

```python
>>> 20.231234 * 3.14159 * 2
>>> (20.231234 ** 2) * 3.14159
```

• With:

```python
>>> radius = 20.231234
>>> pi = 3.14159
>>> circumference = radius * pi * 2
>>> area = (radius ** 2) * pi
```

• Did we lose or gain anything with the new code?
  – Run the two
Using Variables

• When we type in these four lines, Python prints nothing back.

```python
>>> radius = 20.231234
>>> pi = 3.14159
>>> circumference = radius * pi * 2
>>> area = (radius ** 2) * pi
```

• How can we see the actual results?
Using Variables

• When we type in these four lines, Python prints nothing back.

```python
>>> radius = 20.231234
>>> pi = 3.14159
>>> circumference = radius * pi * 2
>>> area = (radius ** 2) * pi
```

• How can we see the actual results?

```python
>>> circumference
127.11648484412
>>> area
1285.8616750694227
```
Doing Repeated Calculations

• What if we want to calculate the area and circumference of circles many times per day (or many times per hour)?

• We can just type in the formulas (as we did in the previous slides) again and again.
  – Any shortcomings of that approach?
The Need for a Program

• What if we want to calculate the area and circumference of circles many times per day (or many times per hour)?

• Typing in the formulas again and again is tedious, and error prone.

• Here is where we can use our first PROGRAM.
Creating a Program

• Create a text file, called “circles.py”.
• From Python shell:
  – File -> New Window
    • Creates a new text window (other than the Shell)
  – File -> Save
    • Allows you to save the file using a name of your choice.
    • IMPORTANT: Make sure you understand what a folder is, and that you know where your file is saved.
    • Talk to the class TA to learn how to do that.
Creating a Program

• Within the file, we put in this text:

```python
# specify the radius value
radius = 25.12

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print("area = ", area)
```
Running the program

• From the text file window, choose Run -> Run Module
(or simply press F5).
Running the program

• From the text file window, choose Run -> Run Module
  (or simply press F5).

Circumference =  157.8334816
area =  1982.388528896
Problem: Radius is Hardcoded

• Why is this a problem?
Problem: Radius is Hardcoded

• Why is this a problem?

• Biggest reason: the user needs to be a programmer.
  – You cannot use this program without changing the program.
Revised Program

• Within the file, we put in this text:

```python
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print("area = ", area)
```
Running the program

• From the text file window, choose Run -> Run Module
(or simply press F5).
Running the program

• From the text file window, choose Run -> Run Module
  (or simply press F5).

Enter the radius of your circle:
Running the program

• From the text file window, choose Run -> Run Module
  (or simply press F5).

Enter the radius of your circle: 5
Running the program

• From the text file window, choose Run -> Run Module
  (or simply press F5).

Enter the radius of your circle: 5
Circumference = 31.4159
area = 78.53975
Running the program

• From the text file window, choose Run -> Run Module
  (or simply press F5).

Enter the radius of your circle: 5
Circumference = 31.4159
area = 78.53975

Is the formula correct? How can you verify?
Testing the program

• Use values 1 and 2 for the radius (also 0):

Enter the radius of your circle: 1
Circumference = 6.28318
area = 3.14159

Enter the radius of your circle: 2
Circumference = 12.56636
area = 12.56636
Testing the program

• Use weird/bad values for the input:
  – strings,
  – Negative numbers
  – Very large numbers
Understanding the Program

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print("area = ", area)

Comment lines:
Are notes to ourselves or other people, the computer ignores all the text from the # sign to the end of the line.
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

Getting user input:
input is a BUILT-IN (PREDEFINED) function in Python. Its job is to print out a message, receive input from the user, and store that input into a string.
Understanding the Program

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

Type conversion:
radius_string is a string, meaning that it is a variable that stores text. Instead, we are interested in the contents of radius_string as a number. The int function is a PREDEFINED Python function, its job is to convert a string into an integer number.
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)
Understanding the Program

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

**Assignments:**
These lines perform numerical calculations, and store the results of those calculations using variables. The variables will exist until the program finishes and they are kept in a table.
Understanding the Program

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

Printing results:
These lines print out results.
print is a predefined Python function. It prints out strings (text), as well as values of variables.

Note: separation by comma.
Changing the code

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

How would you modify this program to print “The area of the circle is “ instead of “area = “?
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print("The area of the circle is ", area)
Changing Variable Names

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

What if I want to change the name of variable radius_string to radius_text?
# get the radius from the user as a string
radius_text = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_text)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

What if I want to change the name of variable radius_string to radius_text?
I have to simply replace ALL occurrences of radius_string with radius_text
Will it change the program behavior?
The Importance of Syntax

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print ("area = ", area)

Python (like all programming languages) is very picky about syntax.
A single misplaced character can make a program not work.
Note the syntax used in this program, and make sure you use the SAME syntax in your code.
Syntax in this Program

# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
# in the beginning of comment lines.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
# The input function takes only one string as argument.
circumference = radius * 2 * pi
print ("Circumference = ", circumference)

# compute and print the area
# The print function takes multiple strings separated by commas.
area = (radius ** 2) * pi
print ("area = ", area)
The Importance of Style

• Original program:

```python
# get the radius from the user as a string
radius_string = input("Enter the radius of your circle: ")

# convert the radius string to an integer.
radius = int(radius_string)

# compute and print the circumference
pi = 3.14159
circumference = radius * 2 * pi
print("Circumference = ", circumference)

# compute and print the area
area = (radius ** 2) * pi
print("area = ", area)
```
The Importance of Style

• Alternative version of the same program:

```python
a = input("Enter the radius of your circle: ")
b = int(a)
c = 3.14159
d = b * 2 * c
print ("Circumference = ", d)
e = b ** 2 * c
print ("area = ", e)
```

• Both versions will run EXACTLY the same.

• What makes the previous version preferable?
The Importance of Style

• Alternative version of the same program:

```python
a = input("Enter the radius of your circle: ")
b = int(a)
c = 3.14159
d = b * 2 * c
print ("Circumference = ", d)
e = b ** 2 * c
print ("area = ", d)
```

• Both versions will run EXACTLY the same.
• What makes the previous version preferable?
• **Readability.** Makes code easier to verify and correct.
The Importance of Style

• Alternative version of the same program:

```python
a = input("Enter the radius of your circle: ")
b = int(a)
c = 3.14159
d = b * 2 * c
print ("Circumference = ", d)
e = b ** 2 * c
print ("area = ", d)
```

• Specific differences:
  – Lack of comments
  – Non-descriptive variable names
  – Lack of empty lines to separate “blocks” of code
Testing and Debugging

• Alternative version of the same program:

```python
a = input("Enter the radius of your circle: ")
b = int(a)
c = 3.14159
d = b * 2 * c
print ("Circumference = ", d)
e = b ** 2 * c
print ("area = ", d)
```

Copy/paste and run this code.
Testing and Debugging

• Run with values 5, and then 1
  – Check the output: notice that both the area and the circumference show the same value, which is incorrect.
Debugging

• Run with values 5, and then 1
  – Check the output: notice that both the area and the circumference show the same value, which is incorrect.

• The error can be due to:
  – Computing (an incorrect result).
  – Printing the incorrect value/variable.
Debugging

• Run with values 5, and then 1
  – Check the output: notice that both the area and the circumference show the same value, which is incorrect.

• The error can be due to:
  – Computing (an incorrect result).
  – Printing the incorrect value/variable.

• The bad style code is harder to check visually.
Some Guidelines

• To learn how to code, you need PRACTICE.
  – What will usually not work:
    • Listen to the lectures.
    • Go and try to do the assignments.
  – What will usually work:
    • Listen to the lectures and KEEP NOTES.
    • Actually type for yourself and run every piece of code that we develop in class. Do this AT HOME, not in class.
    • Understand every line of every piece of code we do in class.
    • Think of variations of what we do in class, and try them.
      – Predict what the variation will do, and test yourself by running it.
    • Then try the assignments.
Some Guidelines

• You need to understand the terminology:
  – Statements, expressions, tokens, literals, functions, strings, variables, operators, ...

• You will encounter many terms in this course. YOU NEED TO LEARN EXACTLY WHAT THEY MEAN.

• DO NOT RELY ON ENGLISH. These terms have meanings in conversational English that are only vaguely related with their meaning in programming.
Terms We Have Seen So Far:

- Command prompt, Shell, IDLE, editor
- Text file
- Filename, file extension
- Folder
- Operator, the order of operators, Python symbols for math operators
- Variable (variable name)
- Literal
- String (in Python)
- Function (in Python, not math)
- Comment
- Input
  - input as a concept and
  - the Python command named input
- Create (a program)
- Run
- Test
- Debug
- Evaluate (and expression)
- Syntax
How to submit your homework

• Windows explorer, location of files, extension of files

• The solution file (that you need to submit for grading) is the file with extension: .py (e.g. circle.py)
  – NOT the run of the IDLE (not the window with ‘>>>’)
  – Pay attention to the difference between IDLE (Shell) and the file EDITOR.

• Submit several files by first making an archive of all of them, and then submitting the archive/zip-file
  – Place all the files in a folder
  – Archive a folder using ZIP and submit the archive (it will have extension .zip).