

CS177 Python Programming

Recitation 1 – All you need to know

Welcome to CS177

Course Instructors

- Prof. Vernon Rego
E-mail: rego@cs.purdue.edu
- Ruby Tahboub (Course Coordinator)
E-mail: rtahboub@purdue.edu
Office: LWSN 2149 #20

GTA Information

[http://courses.cs.purdue.edu/
cs17700:fall15:start](http://courses.cs.purdue.edu/cs17700:fall15:start)

Course Syllabus

- <http://courses.cs.purdue.edu/cs17700:fall15:syllabus>
- Class notes, labs and projects will be posted there.

Course Structure

- Lecture 2 times a week
- Recitation once a week.
- Lab once a week.
- We will take attendance

Textbook

- Python Programming: An Introduction to Computer Science (Second Edition) John Zelle, ISBN 978-1-59028-241-0-9, Franklin, Beedle & Associates Inc., 2004.

Labs

- Weekly lab consists of 3-4 problems
- Lab exercises focus on the material covered the week before
- Sometimes a new concept might be introduced in lab, follow up with your lab TA
- You are encouraged to do the prelab

Projects

- Four projects + Project 0
- You will be writing longer programs than those of labs
- Use language features you learned in class
- Start early!

Grade Distribution

Item	Weight
Midterms	25%
Laboratories	25%
Projects	25%
Final Exam	25%

What do you need?

- Download Python (highly recommended)

<https://www.python.org/download>

- Piazza –Sign up by visiting

[CS17700 on Piazza.](#)

Academic Integrity Policy

- All CS 177 course work must be done individually
- CS 177 uses MOSS tool to verify the integrity of submissions
- In CS 177 a first instance of academic dishonesty will result in a zero for that assignment plus a letter grade deduction at the end of the semester.
- A second instance of academic dishonesty will result in a grade of F.

Objectives

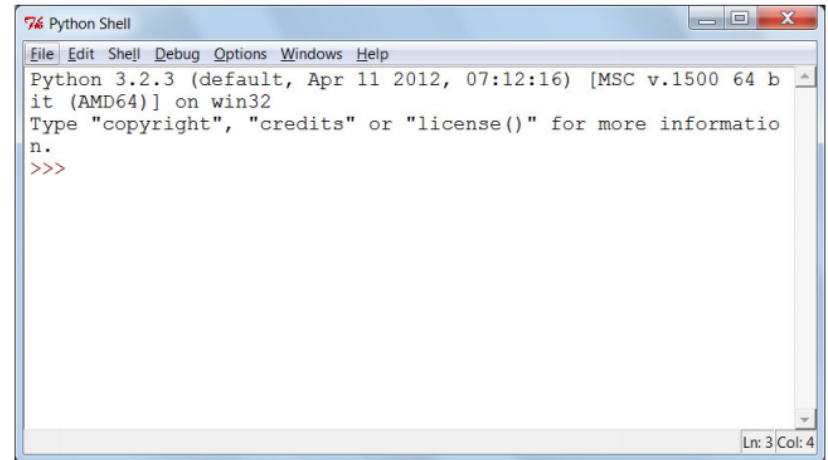
- To learn how to write computer programs.
- To learn how to use the Python programming language.

What is a Computer Program

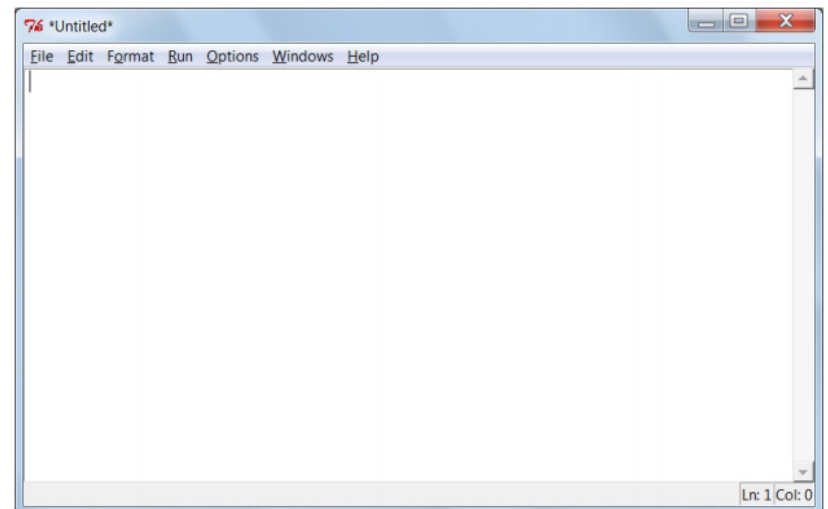
- What is a *computer program*?
 - A detailed, step-by-step set of instructions telling a computer what to do.
 - If we change the program, the computer performs a different set of actions or a different task.
 - The machine stays the same, but the program changes!

Using python

- Go to Start > Search “python”
- Click on IDLE(Python GUI)
- First window opens →
- Click File > New Window
- Second window open →
- Always write code in this window
- Don't forget to save



A screenshot of the Python Shell window. The title bar reads "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area contains the following text: "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.", and a red prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 4".



A screenshot of an "Untitled*" window in the Python IDE. The title bar reads "*Untitled*". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area is empty. The status bar at the bottom right shows "Ln: 1 Col: 0".

What can python do?

- Simple calculations

```
>>>a=5
```

```
>>>b=4
```

```
>>>4=x (WRONG)
```

```
>>>c=a+b
```

```
>>>c
```

```
9
```

- Print Statement

```
>>>print("Hi")
```

```
Hi
```

```
>>>print(c)
```

```
9
```

```
>>>print(a,c)
```

```
5 9
```


Let's write a simple program

```
# by John Smith
```

```
# This program calculates sum of  
# 2 numbers
```

Output

```
>>>4 + 5 = 9
```

```
def main(): ←—————
```

```
    a = 4
```

```
    b = 5
```

```
    c = a + b
```

```
    print(a,"+",b,"=",c)
```

```
main() ←—————
```

main function serves as
an entry point to your
program

invoking main within program,
what happens in case this
statement is omitted?

Reading input from User

- `input` (“prompt message”) is used to read input from user:

```
myInput= input("Type something\n")
print ("This is what you typed:")
print ("You typed:" , myInput)
```

Output:

```
Type something
Welcome to CS177
This is what you typed:
You typed: Welcome to CS177
```

Tip: the type (or representation) of the outcome of `input` is “text”, you need to use function `eval` to convert text to numeric

Reading “numeric” input from User

```
value = input("Type value\n")  
value = eval(value)  
print (value)  
print(value + 10)
```

Output:

```
Type value  
10  
10  
20
```

What happens if eval (i.e., the second line) was omitted?

Project 0

[http://courses.cs.purdue.edu/
cs17700:fall15:projects:project0](http://courses.cs.purdue.edu/cs17700:fall15:projects:project0)

- A Warm up exercise.
- You will exercise writing a python program on your own
- You will learn how to correctly submit your work using turn-in command.
- After Project 0, incorrect submissions will be penalized 5% of grade

Finally ...

- Learning programming is fun and can be useful regardless major.
- IDLE is your friend
- Ask a lot of Questions