# cs002 Lab 6
## *Python Part I*

**Assigned: November 14, 2016** **Due: November 30, 2016**

---

### Introduction

Python is a high-level, object-oriented computing language. It more closely resembles "human language" than "computer language" and utilizes a Graphical User Interface (GUI) to design applications and computer programs. Python is increasing in popularity among employers: knowledge of it will prove useful!

If you are using your own computer, install Python 3.5 from:
https://www.python.org/downloads/.

### Lab Goals

- Gain familiarity with IDLE, the editor and terminal (python shell)
- Understand how to print and store data
- Learn how to model mathematical functions with code

### Lab Assignment

**Hello World**

Open up **IDLE**, a simple editor to write your Python programs in. When you first open IDLE, you will be opened up to a *terminal*. To open up a new file, go to **File -> New File**. A new window should pop up and we will call this your *editor*. This is where you will write your program.

Save this file as *helloWorld.py* by navigating to **File -> Save As** and then save it as *helloWorld*. The .py extension will be added by the editor since you are saving it as a Python file.

In the editor type:

```
print("hello world")
```

Save your file and run the program by going to **Run -> Run Module**. This will run the module in a different window, a *terminal* or the python shell.

Congratulations, you just wrote your first Python program! This program uses the **print** function to print the string "hello world" in the **terminal**.

Here is a list of some basic math operations/commands in python:

# cs002 Lab 6
## *Python Part I*

| Symbol | Operation |
|---|---|
| + | addition |
| - | subtraction |
| * | multiplication |
| / | division |
| ( ) | parenthesis |

The command prompt acts as a simple calculator that allows you to perform these operations. To get started, practice typing simple math equations into the **terminal**. Try to used to all of the operation symbols.

i.e. `>>> 2+2`
(then press enter)

***Note**: the ">>>" do not have to be typed - they should already be there*

```
4
```

**Assignments and Types**

You can assign variable names to different number values. For example, to find the perimeter of a rectangle, the formula is 2*(length * width). If we know the length is 10 and width is 20, we can keep track of these values by assigning variable names to them. Please type the following into the **terminal**:

```
>>> length = 20
```
(Press enter)
```
>>> width = 10
```
(Press enter)
```
>>> perimeter = 2 * (length * width)
```
(Press enter)
```
>>> print(perimeter)
```

perimeter should print 400.

**TASK 1:** To begin this task, go to your **helloWorld.py** file in **IDLE**. (From this point forward, all of today's lab work should be written in this file). On a new line, type:

```
print("Task 1").
```

Once you've done that, please type the above text into the editor to get the answer "400" to appear in the **terminal** by running your *helloWorld.py* file.

**TASK 2:** Your task is to write a program that finds out, on average, how many pizzas Dash can prepare in 90 minutes if he can make 80 pizzas in 2 hours. Once you find that value, print it out to the user. Write your code in your helloWorld.py file and make sure you print "Task 2" before you write your code.

**Hint:** Create a variable called pizzas, and set it equal to an equation that will tell you how many pizzas Dash can make on average (90 minutes is ¾ of 2 hours).

**Lists**

In Python you can make lists of things that you want to group together. A list looks like this:

```
listName = ["pizza", 10, "Bob"]
```

Once assigning your list a name, you can set it equal to a grouping of item inside of square brackets **[ ]**: these can be any combination of strings, numbers, variables, or other lists. Each item in your list **must** be separated by a comma.

**List Indexing**

Each item in a list has an index that allows it to be accessed. List indexing starts at zero.

```
listName = [item1, item2, item3, item4,...]
```

List indices: **0** **1** **2** **3**

To access items in your list you type:

```
listName[indexNumber]
```

where `indexNumber` is one of the list indices

Example:

```
toppings = ["pepperoni", "onion", "spinach", "sausage"]
```

If you want to print "onion", you can:

# cs002 Lab 6
## *Python Part I*

1. Create a variable, 'o' and store onion's name from the list of toppings. Then, print out the variable.

```
o = toppings[1]

print(o)
```

2. Directly print out the index of the list, toppings, that contains the topping you want:

```
print(toppings[1])
```

To change items in your list type:

`listName[indexNumber] = newThing`

where `indexNumber` is the index of the item you want to change to `newThing`

Example:

To change spinach's name to mushroom in your **terminal**, you would type the following:

```
>>> toppings[2] = "mushroom"

>>> print(toppings)
```

*Your new list should now look like this since you changed spinach to mushroom:*

```
["pepperoni", "onion", "mushroom", "sausage"]
```

To add an item to your list use the **append** function: `listname.append(newThing)`

Example:

Add spinach back to the list in your **terminal**:

```
toppings.append("spinach")

print(toppings)
```

*Your list should now look like this:*

```
["pepperoni", "onion", "mushroom", "sausage", "spinach"]
```

**Input**

You can get direct input from your user by using the input function. The input function is set up like this:

```
myVar = input("Question/statement that you want to appear to the user")
```

In the **terminal** type:

```
>>> name = input("Hi, what is your name?")
```

You will then be prompted to input a name. Type your name in quotes.

```
>>> Hi, what is your name?"Dash"
```

Next, in the **terminal** type:

```
>>> print("hello " + name)
```

In this example, name will represent whatever you type into the shell after the question *"Hi, what is your name?"* Note that whatever the user has given you will be saved as a string. If you want the user to input a number, you will have to cast the input as an integer by typing: **`int(what you want to turn into an int)`**

Example:

```
age = int(input("How old are you"))
```

Now you can do math manipulations and other integer things with your variable "age".

Example:

```
print(age+5)
```

# cs002 Lab 6
## *Python Part I*

**Conditionals**

Conditional expressions use logic to determine which path of action to take based upon conditions set by the programmer. Conditional statements are often set up in the following way:

```
if (condition):
        (statements)
elif (condition):
        (statements)
else:
        (statements)
```

There are a few critical things to remember when writing if statements:

1. You must follow each if-statement with a colon, or you will get a syntax error.
2. Indentation must be correct.
    a. Each condition must be aligned with one another, and the statements following them must be indented one tab over.
3. The conditions that follow if and elif must evaluate to true or false.
4. The conditions of the if and elif statements must be mutually exclusive (they cannot both be true), or your program probably won't act the way you want it to.

The following is a list of conditional operators:

| **Symbol** | **Operation** |
|---|---|
| < | Less Than |
| <= | Less Than Or Equal |
| > | Greater Than |
| >= | Greater Than Or Equal |
| == | Equal |
| and | And |
| or | Or |
| not | Not |

The "or" and "and" operators are used to connect simple conditionals together. If the first OR the second conditional within the larger conditional is true, then the entire statement is true. If the first AND the second conditional is true within the larger conditional, then the entire statement is true. Lastly, the "not" is a logical negation. If the conditional following a "not" statement is false, then the entire statement is true.

# cs002 Lab 6
## *Python Part I*

### TASK 3:

Using what you learned about the input function and conditionals, create a simple program in the **editor** (not terminal) that asks the user to enter how many pizzas they want. Based on the value of the integer entered, several things can happen:

1. If the number of pizzas is less than or equal to 0, print out "No pizzas"

2. If the number is greater than 0 but less than 5, print out "between 0 and 5 pizzas", followed by the actual number of pizzas entered.

3. If the number is greater than 5, print out "More than 5 pizzas", followed by the actual number of pizzas entered.

### Check-off Requirement

Please show your TA the following:

1. Task 1: Print the perimeter of a rectangle
2. Task 2: Print how many pizzas Bob can make in 90 minutes
3. Task 3: Print out how many pizzas the user inputs and the corresponding message

### Submission

To submit this lab, please raise your hand so a TA can come check your work. Make sure you tell the TA to check you off their list, or else you will not receive credit. If you do not finish your lab in the allotted time, please either come to office hours or another lab section to finish your work. If you are unable to complete this lab due to sickness or injury, please contact BOTH Don and the cs002 TAs via email. Please contact the cs002 TA's if you have any further questions.

**cs002 TA Email: cs002tas@cs.brown.edu**
**Don Stanford's Email: don.stanford@gmail.com**