LAB #3
Programming w/ Python

1. In this course, all our labs involve paired programming. You do not have to keep the same partner for each lab, but you MUST work with someone in each lab, as specified in the student handout.

2. At this time, you need to pair with someone in the lab, and finish the rest of the lab as a pair.

Play Robozzle Games (5 pts)

3. Go to the robozzle website: http://robozzle.com/. Go to the Campaign tab at the top. You will try to play zipline, Two stripes, and mountains for the first hour of the lab. Do not take more than an hour, and write down (or take a screen shot) of the instructions that solve each of the puzzles.

Begin Programming in Python (3 pts)

4. To begin, we are going to execute a few python statements to understand the semantics and syntax before writing a full program. Let’s enter the following examples for you to see what python syntax is. First, Python statements do not need to be ended by a semicolon, which is ignored if supplied. However, it might be a good habit to get into because other languages require it, and it will be required in CS 161! Let’s start the Python interpreter by just typing python3 at the command prompt.

```
$ python3
Python 3.2.5 (default, Jul 21 2014, 15:49:22)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```
• **Print information to the screen** for the user to read. Strings of text have to be surrounded by quotes.
  ```python
  print("Hello Everyone");
  ```

• **Assign information to variables** that store the information in memory to retrieve for later use. You can assign strings of text, whole numbers, and real numbers.
  ```python
  real_num= 19.4;
  whole_num = 250;
  message = "Hello Everyone!";
  ```

Now, print the values in these variables by using the variable name.
```python
print(real_num);
pint(whole_num);
print(message);
```

There are expressions that you can use to **change the values in the variables**. Let’s change the values in the variables we just created.
```python
real_num = real_num + 10.6;
whole_num = whole_num + 10;
message = message + "Python!";
```

Now, **print the values in these variables again** by using the variable name.
```python
print(real_num);
pint(whole_num);
print(message);
```

Other **math operators** include subtraction, division, multiplication, exponent, and mod/remainder:
```python
print(10 – 5);
print(10 / 5);
print(10 * 5);
print(10 ** 5);
print(10 % 5);
```

What if we want to **concatenate variables/information together**? We have to convert the numbers to strings to put them together with strings in a print statement. This is because the + symbol can also be used between numbers to mean addition. It depends on the context you are using the + symbol. We can temporarily change the type of a variable using typecasting. In this case we need the numbers to act as strings of text. We use a function called `str()` to do this.
```python
print(str(real_num)+" "+str(whole_num)+" "+message+" pretty cool!");
print(real_num+whole_num);
```
• Next, we need to be able to **get information from the user**. Since we are reading something from the user, we need to be able to store it somewhere to be able to retrieve it later. Problem is that ALL the input comes in as strings of text, not a number. We can prevent this with a typecast to go from a string of text to a different kind of data, such as a real or whole number. In CS, we call these floating point and integer numbers, respectively. Therefore, we have to use the functions float() and int() to change the string to a real number or whole number.

```
real_num = input("Enter a different real number: ");
print(real_num);
print(real_num+message);
```

```
real_num = float(input("Enter another number: "));
print(real_num);
print(str(real_num)+message);
```

To exit, just type `exit();`

• Let’s put it all together in an example program. Open a new file by typing:

```
vim lab3.py
```

```
num = input("Enter a number: ");
print("Your number is: " + num);

num = int(num)**10;
print("Your number to the 10th exponent: " + str(num));

num = int(input("Enter a different num: "));
num = num**10;
print("Your new number to the 10th exponent: " + str(num));
```

Now run your program by typing `python3 lab3.py`.

**Design a Program on Paper (2 pts)**

5. Now, **write the steps needed to solve the following problem** on a piece of paper:

The user provides the amount of coupons won as input and a program outputs how many candy bars and gumballs you can get. Remember, you can redeem 10 coupons for a candy bar or 3 coupons for a gumball, and you prefer candy bars to gumballs. Output how many candy bars and gumballs you get, when you spend all of your coupons on candy bars first and any remaining coupons on gumballs.

**Make sure you sign-up with a TA for demoing/explaining your Assignment #2 next week. This is how all assignments are graded in the course, and if you sign-up and do not make your appointment without rescheduling, then you will be penalized 50 points!!!**