

1 Introduction

This worksheet will serve as a brief review of everything that has occurred in CS61A over the last few weeks.

1.1 Functions

Functions, in the most basic sense, are a block of code, that can be used to perform some type of action. Functions have multiple uses, and can be used as return values, nested definitions, and arguments.

The structure

```
1 def hello(hi):  
2     ...  
3     return bye
```

In the above function, "hello" is the function name, hi is the argument and bye is the return value. Now that we've gone over the basics of functions, let's describe all of their use cases.

1.1.1 Functions as arguments

Functions can be used as arguments. Let's take the following example:

```
1 def eatFood(makeFood):  
2     makeFood()  
3     print("I have eaten!")
```

Here we have a function *eatFood*. In this, we have an argument, *makeFood*. We call the function *makeFood* and then say that we have eaten. Depending on the food that we want to eat, we will have a different method to make it. However, regardless of how we make the food, we will eat it the same way, with our mouths! So instead of making a new function for eating every type of food, we have a function for eating that takes in a function that makes our food, allowing us to make a more universal function.

2 Butter Nutter

(a) Label the parts of the following function:

```
1 peanut = butter = "crunchy"  
2 jelly = "strawberry"  
3  
4  
5 def sandwich(butter, bread):
```

```
6     if (len(bread) > len(butter) and len(butter) != 0):
7         return sandwich(butter[1:], bread + butter[0])
8     return bread
9
10
11 sandwich(peanut + " ", jelly)
```

Solution:

```
1 peanut = butter = "crunchy" #global variables
2 jelly = "strawberry" #global variable
3 def sandwich(butter, bread): #function name
4     if (len(bread) > len(butter) and len(butter) != 0): #conditional
5         return sandwich(butter[1:], bread + butter[0]) #recursive call
6     return bread #return statement
7
8
9 sandwich(peanut + " ", jelly) #function call
```

(b) What does the above code snippet return?

Solution: strawberrycrunchy