

# Homework1

January 14, 2015

## 1 Finance 4480

### 1.1 Homework 1

**Due January 23, 2015**

This homework is all about putting to use your knowledge of **Python** from the different building blocks that you have seen so far.

We will be talking about these problems and I will give lots of hints in class!

#### 1.1.1 Problem 1

This problem is known as the *chicken nuggets* problem. It goes like this: you walk into Chick Fil-A with an unlimited amount of money (and appetite!). You can purchase nuggets in containers of 6, 9, and 20.

Write a program to tell you the **highest** number of nuggets that you **cannot** purchase! Re-read that just in case it went past you the first time! The highest number that you cannot get. For example, you can get 15 nuggets by purchasing and box of 6 and a box of 9 nuggets. You can get 18 by purchasing 2 boxes of 9 nuggets. But with no combination of 6, 9, and 20 can you purchase 17 nuggets. What is the highest number that you cannot get?!

This simple game will give you experience assembling different bits of **Python** programming to find the solution. It will also employ a very simple numerical method called *brute force* search.

Please write your solution in the code cell below:

```
In [13]: def main():
          # your code goes here!
          print("Good luck!") # remove this line of code for your final solution!

          if __name__ == "__main__":
              main()
```

Good luck!

#### 1.1.2 Problem 2

In the book [Python Programming for the Absolute Beginner, 3rd Edition](#) the author teaches **Python** through some simple game programming. One of the first games that he shows how to write is the so-called **Guess My Number** game, which is the children's game of guessing some one's secret number (a number between 1 and 100).

An implementation of the game in **Python** might look something like this:

```
In [16]: import random

          def print_header():
              print("\tWelcome to 'Guess My Number'!")
              print("\tI'm thinking of a number between 1 and 100.")
```

```

    print("\tTry to guess it in as few attempts as possible.\n")

def print_footer(the_number, tries):
    print("You guessed it! The number was", the_number)
    print("And it only took you", tries, "tries!\n")
    print("\n\nPress the enter key to exit.")

def main():
    # print the greeting banner
    print_header()

    # set the initial values
    the_number = random.randint(1, 100)
    guess = int(input("Take a guess: "))
    tries = 1

    # the game loop
    while guess != the_number:
        if guess > the_number:
            print("Lower ...")
        else:
            print("Higher...")

        guess = int(input("Take a guess: "))
        tries += 1

    print_footer(the_number, tries)

if __name__ == "__main__":
    main()

```

```

Welcome to 'Guess My Number'!
I'm thinking of a number between 1 and 100.
Try to guess it in as few attempts as possible.

```

```

Take a guess: 50
Lower ...
Take a guess: 25
Lower ...
Take a guess: 12
Higher...
Take a guess: 18
Lower ...
Take a guess: 15
Lower ...
Take a guess: 13
You guessed it! The number was 13
And it only took you 6 tries!

```

```

Press the enter key to exit.

```

Make sure you understand each bit of the above program! Study it, play with it, pick it apart, re-write on your own computer or on SMC.

Your task in this problem is to now write a version of the *Guess My Number* game where you and the computer switch roles! That is right: you think of a number and the computer must guess it in as few attempts as possible. You will need to encode your guessing logic to the program solution.

This might seem like silly game play, but in order to solve the problem you must use an algorithm called **binary search** or the **bisection method** to solve the problem correctly. This is our first attempt at programming a simple algorithm. We will see this algorithm later in the context of the ***Black-Scholes-Merton Option Pricing Model*** to calculate the implied volatility of the model. This is something that options traders do thousands and thousands of times a day!

Please use the code cell below to write your solution:

```
In [14]: def main():
          # your code goes here!
          print("Good luck!") # delete this line of code

          if __name__ == "__main__":
              main()
```

Good luck!

In []: