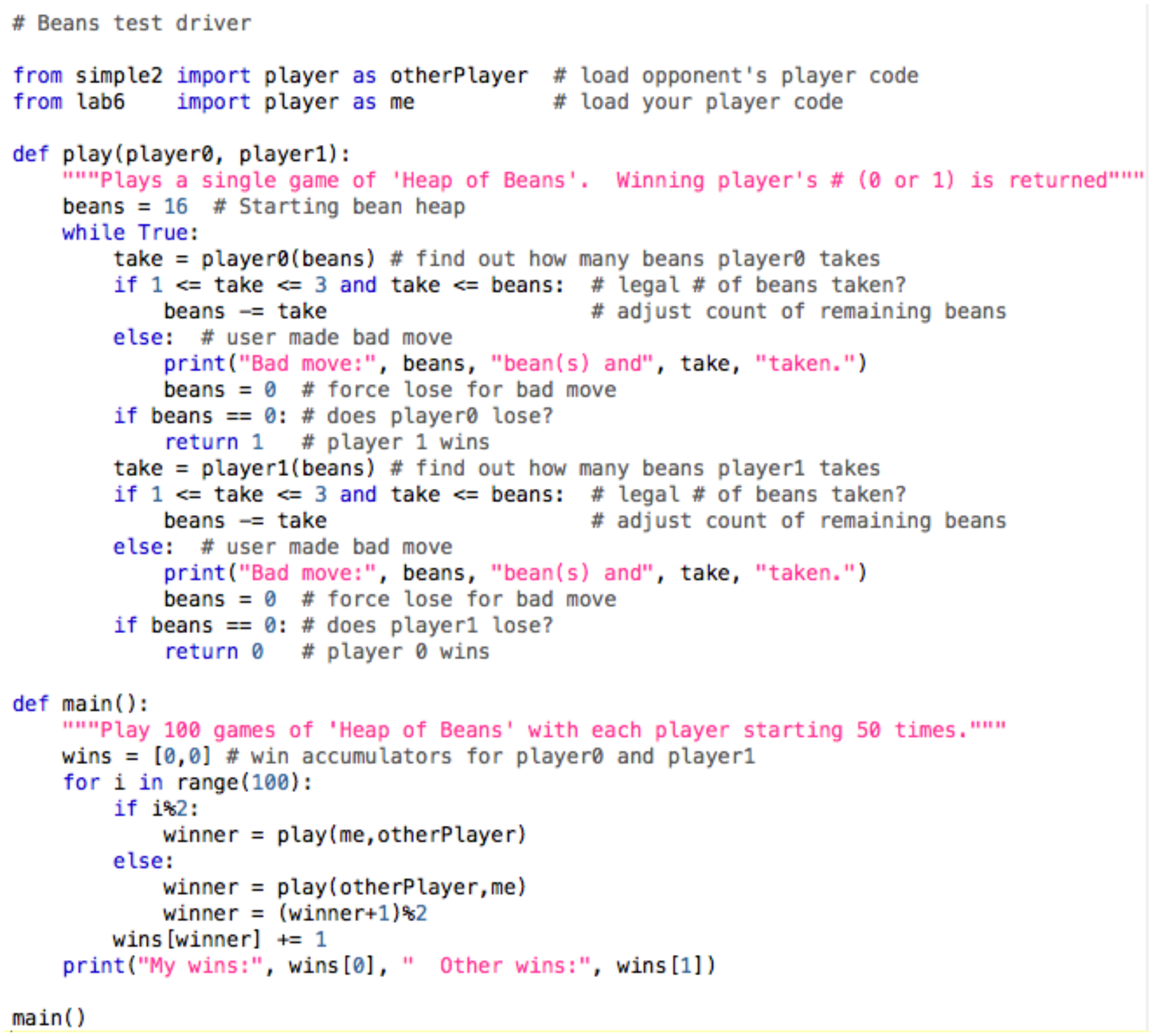
IT 210 – Fundamentals of Programming – Lab 6

You will be working as an individual on this lab and build a player (a function that you write) to play Heap of Beans. This game is played with two players who take turns removing beans from a heap of beans on the table. Each player in turn is permitted to remove 1, 2, or 3 beans for a turn. The player who takes the last bean loses. Your goal is to write the best function (producing the largest number of wins) to play the part of one player in this game. Your function must be named **player** and expect one argument, the number of beans currently remaining in the heap. It will return the number of beans (an integer) to be removed from the heap this turn. Its basic structure is like this:

```
## Play "Heap of Beans".  Use the best strategy you can devise to beat an opponent.
#  1, 2, or 3 beans can be removed in a turn, but never more than the number of beans
#  currently on the table.
# @param beans – the number of beans currently on the table
# @return – the number of beans to be removed this turn
#
def player(beans):
      .
      .
      .
      return beansToTake
```

I have provided a zip file on D2L that contains 3 files to help you to test your program: beans.py is the main testing program (shown below) that will allow you to compete against other programs (including yourself if you want) and two very simple programs that are very poor competitors and that you should usually beat. The test program uses a few python features you haven't seen yet, but you should get a pretty good idea of how it works by reading it. The program I use for testing is similar, but will automatically match you with other competitors



```
# Beans test driver

from simple2 import player as otherPlayer  # load opponent's player code
from lab6    import player as me           # load your player code

def play(player0, player1):
    """Plays a single game of 'Heap of Beans'.  Winning player's # (0 or 1) is returned"""
    beans = 16  # Starting bean heap
    while True:
        take = player0(beans) # find out how many beans player0 takes
        if 1 <= take <= 3 and take <= beans:  # legal # of beans taken?
            beans -= take                     # adjust count of remaining beans
        else:  # user made bad move
            print("Bad move:", beans, "bean(s) and", take, "taken.")
            beans = 0  # force lose for bad move
        if beans == 0: # does player0 lose?
            return 1   # player 1 wins
        take = player1(beans) # find out how many beans player1 takes
        if 1 <= take <= 3 and take <= beans:  # legal # of beans taken?
            beans -= take                     # adjust count of remaining beans
        else:  # user made bad move
            print("Bad move:", beans, "bean(s) and", take, "taken.")
            beans = 0  # force lose for bad move
        if beans == 0: # does player1 lose?
            return 0   # player 0 wins

def main():
    """Play 100 games of 'Heap of Beans' with each player starting 50 times."""
    wins = [0,0] # win accumulators for player0 and player1
    for i in range(100):
        if i%2:
            winner = play(me,otherPlayer)
        else:
            winner = play(otherPlayer,me)
            winner = (winner+1)%2
        wins[winner] += 1
    print("My wins:", wins[0], "  Other wins:", wins[1])

main()
```

Your program, the test driver and the competitor programs should all be in the same folder. To use the test driver, on line 4, change lab6 to the name of your file (without the .py). On line 3 select simple1, simple2 or your file name as the competitor.

Submit your lab to the D2L dropbox not later than 11:59 pm Friday evening. Submit only the file with your best player and name your file Lab6.py. I will have your program compete against each student who submits a conforming function (one that does not violate the rule of the game and only returns valid responses) and you will receive a grade based on your total number of wins.