Written Response

2a.

The program is written in the Python programming language. The purpose of the program is to simulate a 15-point round of saber fencing via a text-based interface. In the video, I run the program, score three points, then I let the computer score two points.

2b.

Originally, I had created the defense function of the program, wherein the player simply blocked the opponent's attacks. From there, I added an attack function and the player could strike back at the opponent. I encountered a bug that would not allow the program to initialize the function *strike()*. It was only with the inspiration from Devon that I was able to gain a new perspective and crush the bug. From there, I added a score count function. For a while, the score count function would not display, no matter how much I tweaked it. I needed to change the display function from *return()* to *print()*.

2c.

```
diffchoice=1
scoreboard1=0
scoreboard2=0


#This is the function that both displays score and determines if play resumes.
def scorecheck(scoreboard1, scoreboard2):
    print "Your score: ",scoreboard1," ,Opponent's score: ", scoreboard2
    if scoreboard1<15 and scoreboard2<15:

        attack()
    else:
        if scoreboard1==15:
            print "You win the bout."
        elif scoreboard2==15:
            print "Your opponent wins the bout."

#This will determine the chance of whether the opponent will successfully block your attack.
def strike(move,diffchoice):
    diff=int(diffchoice)
    def1=randint(0,diff)
    atck1=int(move)
    if atck1==4:
        print "You attack your opponent's flank"
        if def1==0:
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1

 #
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent blocks your blade."
            attack()


    elif atck1==3:
        print "You attack your opponent's mask"
        if def1==0:
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)

        else:
            print "Your opponent blocks your blade."
            attack()


    elif atck1==2:
        print "You attack your opponent's chest"
        if def1==0:
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent blocks your blade."
            attack()

```

This section of code was the most difficult to code. Here, I had to first think about how a program might block an opponent's attack, so I had to create multiple boolean statements to determine the program's course of action in response to the action of the player, so I created a pseudorandom number generator (0,1) which determined if the program managed to block the attack or not. From there, I had to tack on the variable of difficulty, so I inserted a variable within the pseudorandom number generator, and attached that variable to a prompt that existed when the "fencingsim" program was first initialized. In addition, I had to tack on an additional function that functioned as a scoreboard, which required the creation of two global variables (i.e. "scoreboard1" and "scoreboard2"). At some point, I could not get the *scoreboard* function to function (pun not intended), so the position of the *scoreboard* function had to be changed in order to function correctly.

2d.

```
#This will determine the result of your defensive choice
def defense(atck, reagonac):
    act = int(reagonac)
    #If the opponent chooses not to move.
    if atck==0:
        if act==0:
            print "Nobody moves.  The silence becomes awkward quickly."
            attack()
        elif act>=4:
            print "You strike your opponent without the opponent offering an effort to block. \nYou are dumbfounded at your opponent's inaction."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "You move your blade to block, but nothing came. Your opponent takes action."
            attack()
    #If the opponent chooses to strike.
    else:
        if act==atck:
            print "You blocked your opponent. Your move."
            move = raw_input()
            strike(move,diffchoice)
        elif act>=4:
            print "Your blade hits your opponent, but the opponent's blade touches you first.  The opponent scores."
            global scoreboard2
            scoreboard2+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent's blade makes contact. Your opponent scores."
            global scoreboard2
            scoreboard2+=1
#
            scorecheck(scoreboard1,scoreboard2)
```

In order to create a “difficulty setting”, I created a separate variable, “diffchoice”, and used that variable as the denominator of the fraction that determines the probability of the opponent missing the target.

3.

```python
from random import randint

diffchoice=1
scoreboard1=0
scoreboard2=0


#This is the function that both displays score and determines if play resumes.
def scorecheck(scoreboard1, scoreboard2):
    print "Your score: ",scoreboard1,", Opponent's score: ", scoreboard2
    if scoreboard1<15 and scoreboard2<15:

        attack()
    else:
        if scoreboard1==15:
            print "You win the bout."
        elif scoreboard2==15:
            print "Your opponent wins the bout."

#This will determine the chance of whether the opponent will successfully block your attack.
def strike(move,diffchoice):
    diffpoints(diffchoice)
    def1=randint(0,difp)
    attack=randint(0,3)
    if attack==1:
        print "You attack your opponent's flank"
        if(def1==0):
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1

#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent blocks your blade."
            attack()


    elif attack==2:
        print "You attack your opponent's mask"
        if def1==0:
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)

        else:
            print "Your opponent blocks your blade."
            attack()

    elif attack==3:
        print "You attack your opponent's chest"
        if def1==0:
            print "Your opponent fails to block your blade. \nYou score a point."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent blocks your blade."
            attack()


#This will determine the results of your defensive choice
def defense(atak, response):
    ans = int(response)
    #If the opponent chooses not to move.
    if atak==0:
        if ans==0:
            print "Nobody moves.  The silence becomes awkward quickly."
            attack()
        elif ans==1:
            print "You strike your opponent without the opponent offering an effort to block. \nYou are dumbfounded at your opponent's inaction."
            global scoreboard1
            scoreboard1+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "You move your blade to block, but nothing came. Your opponent takes action."
            attack()
    #If the opponent chooses to strike.
    else:
        if ans==atak:
            print "You blocked your opponent. Your move."
            move = raw_input()
            strike(move,diffchoice)
        elif ans!=0:
            print "Your blade hits your opponent, but the opponent's blade touches you first.  The opponent scores."
            global scoreboard2
            scoreboard2+=1
#
            scorecheck(scoreboard1,scoreboard2)
        else:
            print "Your opponent's blade makes contact. Your opponent scores."
            global scoreboard2
            scoreboard2+=1
#
            scorecheck(scoreboard1,scoreboard2)

#This selects the action of the NPC opponent.
def attack():
    atak = randint(0,3)
    if atak==0:
        response = raw_input ("The opponent does not move. What is your response?\n")

    elif atak==1:
        response = raw_input ("The opponent moves his blade towards your flank.  What is your response?\n")

    elif atak==2:
        response = raw_input ("The opponent moves his blade towards your mask.  What is your response?\n")

    elif atak==3:
        response = raw_input ("The opponent moves his blade towards your chest. What is your response?\n")

    defense(atak, response)

#This is where the instructions for the program will be given.
def fencingain():
    print "Type 0 to do nothing.\nType 1 to protect your flank.\nType 2 to protect your mask.\nType 3 to protect your chest.\nType 4 to attack your opponent's flank.\nType 5 to attack your opponent's mask.\nType 6 to attack your opponent's chest."
    diffchoice=int(raw_input("What is your difficulty selection?\nEnter 1 for easy. Enter 2 for medium. Enter 3 for hard.\n"))
    print "The opponent makes the first move."
    attack()


#This will initialize the program
fencingain()
```