

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.

```
1 diffchoice=1
2 scoreboard1=0
3 scoreboard2=0
4
5
6
7
8
9
10 #This is the function that both displays score and determines if play resumes.
11 def scorecheck(scoreboard1, scoreboard2):
12     print "Your score: ",scoreboard1," ,Opponent's score: ", scoreboard2
13     if scoreboard1<15 and scoreboard2<15:
14         attack()
15     else:
16         if scoreboard1==15:
17             print "You win the bout."
18         elif scoreboard2==15:
19             print "Your opponent wins the bout."
20
21 #This will determine the chance of whether the opponent will successfully block your attack.
22 def attack(move,diffchoice):
23     diff=int(diffchoice)
24     diff=randint(0,diff)
25     attack=int(move)
26     if attack==4:
27         print "You attack your opponent's flank"
28         if diff==0:
29             print "Your opponent fails to block your blade. \nYou score a point."
30             global scoreboard1
31             scoreboard1+=1
32
33     #
34     scorecheck(scoreboard1,scoreboard2)
35     elif:
36         print "Your opponent blocks your blade."
37         attack()
38
39
40     elif attack==5:
41         print "You attack your opponent's mask"
42         if diff==0:
43             print "Your opponent fails to block your blade. \nYou score a point."
44             global scoreboard1
45             scoreboard1+=1
46     #
47     scorecheck(scoreboard1,scoreboard2)
48
49     elif:
50         print "Your opponent blocks your blade."
51         attack()
52
53
54     elif attack==6:
55         print "You attack your opponent's chest"
56         if diff==0:
57             print "Your opponent fails to block your blade. \nYou score a point."
58             global scoreboard1
59             scoreboard1+=1
60     #
61     scorecheck(scoreboard1,scoreboard2)
62
63     elif:
64         print "Your opponent blocks your blade."
65         attack()
66
```

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.

```

66
67
68 #This will determine the result of your defensive choice
69 def defense(attack, response):
70     act = int(response)
71     #If the opponent chooses not to move.
72     if attack==0:
73         if act==0:
74             print "Nobody moves. The silence becomes awkward quickly."
75             attack()
76         elif act==4:
77             print "You strike your opponent without the opponent offering an effort to block. \nYou are dumbfounded at your opponent's inaction."
78             global scoreboard1
79             scoreboard1+=1
80 #
81         scorecheck(scoreboard1, scoreboard2)
82     else:
83         print "You move your blade to block, but nothing came. Your opponent takes action."
84         attack()
85 #If the opponent chooses to strike.
86 else:
87     if act==attack:
88         print "You blocked your opponent. Your move."
89         move = raw_input()
90         strike(move, diffchoice)
91     elif act==4:
92         print "Your blade hits your opponent, but the opponent's blade touches you first. The opponent scores."
93         global scoreboard2
94         scoreboard2+=1
95 #
96         scorecheck(scoreboard1, scoreboard2)
97     else:
98         print "Your opponent's blade makes contact. Your opponent scores."
99         global scoreboard2
100         scoreboard2+=1
101 #
102         scorecheck(scoreboard1, scoreboard2)
103

```

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.

```

1 from random import randint
2
3 difficulty=1
4 score=0
5 score=0
6
7
8
9
10 #this is the function that both displays score and determines if play resumes.
11 def scorecheck(scoreboard1, scoreboard2):
12     print "You scored: ", scoreboard1, " Opponent's score: ", scoreboard2
13     if scoreboard1<10 and scoreboard2<10:
14
15         attack()
16
17     elif scoreboard1==10:
18         print "You win the bout."
19     elif scoreboard2==10:
20         print "Your opponent wins the bout."
21
22 #this will determine the chances of whether the opponent will successfully block your attack.
23 def successchance(difficulty):
24     r=randint(0,100)
25     scorecheck(scoreboard1, scoreboard2)
26     if r<=difficulty:
27         print "You attack your opponent's flank"
28         if r<=10:
29             print "Your opponent fails to block your blade. You score a point."
30             global scoreboard1
31             scoreboard1+=1
32
33 #
34     scorecheck(scoreboard1, scoreboard2)
35     elif r>10:
36         print "Your opponent blocks your blade."
37         attack()
38
39 #
40     elif r>10:
41         print "You attack your opponent's neck"
42         if r<=10:
43             print "Your opponent fails to block your blade. You score a point."
44             global scoreboard1
45             scoreboard1+=1
46
47 #
48     scorecheck(scoreboard1, scoreboard2)
49     elif r>10:
50         print "Your opponent blocks your blade."
51         attack()
52
53 #
54     elif r>10:
55         print "You attack your opponent's chest"
56         if r<=10:
57             print "Your opponent fails to block your blade. You score a point."
58             global scoreboard1
59             scoreboard1+=1
60
61 #
62     scorecheck(scoreboard1, scoreboard2)
63     elif r>10:
64         print "Your opponent blocks your blade."
65         attack()
66
67
68 #this will determine the results of your defensive choice
69 def defence(choice, response):
70     r=randint(0,100)
71     if r<=10:
72         print "You choose not to move."
73         if r<=10:
74             print "Your opponent's blade strikes you. The opponent scores a point."
75             global scoreboard2
76             scoreboard2+=1
77         elif r>10:
78             print "You strike your opponent without the opponent offering an effort to block. You are dumbfounded as your opponent's blade strikes."
79             global scoreboard2
80             scoreboard2+=1
81
82 #
83     scorecheck(scoreboard1, scoreboard2)
84     elif r>10:
85         print "You move your blade to block, but nothing came. Your opponent takes action."
86         attack()
87
88 #if the opponent chooses to strike.
89     elif r>10:
90         if r<=10:
91             print "You blocked your opponent. Your move."
92             move = raw_input()
93             if move=="diffensive":
94                 print "Your blade hits your opponent, but the opponent's blade touches you first. The opponent scores."
95                 global scoreboard2
96                 scoreboard2+=1
97
98 #
99     scorecheck(scoreboard1, scoreboard2)
100     elif r>10:
101         print "Your opponent's blade makes contact. Your opponent scores."
102         global scoreboard2
103         scoreboard2+=1
104
105 #
106     scorecheck(scoreboard1, scoreboard2)
107
108 #this selects the action of the bot opponent.
109 def attack():
110     move = randint(0,10)
111     if move==0:
112         response = raw_input("The opponent does not move. What is your response?")
113
114     elif move==1:
115         response = raw_input("The opponent moves his blade towards your flank. What is your response?")
116
117     elif move==2:
118         response = raw_input("The opponent moves his blade towards your neck. What is your response?")
119
120     elif move==3:
121         response = raw_input("The opponent moves his blade towards your chest. What is your response?")
122
123     defence(move, response)
124
125 #this is where the instructions for the program will be given.
126 def fusingain():
127     print "Type 0 to do nothing, Type 1 to protect your flank, Type 2 to protect your neck, Type 3 to protect your chest, Type 4 to attack your opponent's flank, Type 5 to attack your opponent's neck, Type 6 to attack your opponent's chest."
128     difficulty=raw_input("What is your difficulty selection? Enter 1 for easy, Enter 2 for medium, Enter 3 for hard.")
129     print "The opponent makes the first move."
130     attack()
131
132 #this will initialize the program
133 fusingain()

```