

November 2021
Westport, CT

William McGonagle
Nina Yuen
Greens Farms Academy



Fairfield Programming Association

Rules and Regulations 2022

Edition 1

<https://fairfieldprogramming.org/>

Introduction

To Whom It May Concern,

I am building a league of schools to host competitions testing merit in programming. It will be known as the Fairfield Programming Association (FPA). To create this organization, I only need a few things: support from your institution, two student delegates from your institution, and your help maintaining the cohesiveness of the FPA during its adolescent years. These first two years are likely going to be the most challenging for this new organization, and the FPA will be tested at its very foundation. That is why I need your help, and I have outlined what you can do below.

Firstly, your institution needs to support the FPA. This is not in any financial sense, but rather in a mutually beneficial partnership. The FPA's competitions would allow your students to create connections with students in other schools and it would give them the opportunity to thrive in a competitive environment- all while learning. All your institution would need to support the FPA is the commitment to hosting at least one competition each year. The logistics of these competitions are outlined in the below document.

You will need to elect two student delegates from your institution to help host the competitions. These delegates will be responsible for meeting in a committee with the other FPA delegates before each competition to design them so they are fair, challenging, and most importantly, fun. The logistics of the delegates and committee meetings will be outline in the below document.

Finally, your job will be to listen for any communications that the FPA sends out, and corral your students into teams for each competition. It is expected that every institution attends at least two other competitions, but that information, again, is outlined in the below document.

Best Regards,
William McGonagle

A handwritten signature in black ink, appearing to read 'WM', with a horizontal line extending from the end of the signature.

Table of Contents

Introduction	1
Table of Contents	2
Team Rules	4
Team Names	4
Team Coaches	4
Coach Attendance	4
Replacement Coaches	4
Language and Professionalism	4
Allowed and Disallowed Language	4
Put-Downs and Insults	5
Respect to Delegates	5
Respect to Coaches	5
Prohibited Substances	5
Alcohol Products	6
Marijuana and Narcotic Items	6
Tobacco and Nicotine Products	6
The Ferragu Clause	6
Institution Suspension	6
Competition Suspension	6
Three Month Suspension	6
One Year Suspension	6
Two Year Suspension	7
Delegate Rules	8
Number of Delegates	8
Meeting Times	8
Chief Delegate	8
Chief Delegate Responsibilities	8
Chief Delegate Voting	8
Speaking Times	8
Contest Rules	9
Problem Rules	10
Points and Scoring	10

Time Spent Scoring	10
Difficulty Based Scoring	10
Style Guide	11
Problem Input and Output	11
Problem Accessibility	11
Solution Formatting	11
Solution Design	11
Solution Input and Output	11
Libraries and Imports	12
Problem Length	12
Allowed Methods in the Standard Library	12
Allowed Methods in the Math Library	12
Example Problems	14
Example 1 - Music Mixup	14
Intro	14
Input and Output Testing	15
Sample Solution	15
Example 2	16
Intro	16
Input and Output Testing	16
Sample Solution	17

Team Rules

Team Names

Teams can be referred to either as their school name or their school mascot. If the committee decides through a majority vote that a team mascot is not appropriate or is offensive, the team will only be referred to by their school name. Team names cannot contain vulgar, or forbidden language (see Language and Professionalism).

Team Coaches

Teams are required to have at least one coach. This individual must be over the age of 18 and employed by the institution that the team represents. The coach has legal responsibility of the students of their institution while attending the competition.

Coach Attendance

The coach, or a replacement coach, needs to attend every competition that the team attends. If a team does not have a coach present at the competition, the team is disqualified from that competition.

Replacement Coaches

Replacement coaches are allowed but greatly discouraged. A replacement coach must meet the same criteria as a regular coach.

Language and Professionalism

All members of the Fairfield Programming Association, Delegates, Coaches, Competitors, and even parents, are expected to be professional and adhere to the following guidelines for respectful and kind actions.

Allowed and Disallowed Language

Words can be split into three categories of appropriateness: appropriate words, vulgar words, and forbidden words. The first category of words is allowed in competition, but the other two are punishable.

Appropriate words are words that are used in everyday conversation. If you would be willing to say these words to your grandmother, to your boss, in a church, or to a child, they are likely to be appropriate. Swears, sexual language and slurs are not in the category of appropriate words.

Vulgar words are words that are not used in everyday conversation. These are mostly swears and sexual language. Phrases that can be deemed as sexual language or swear-like also fall into the

category of vulgar words. Usage of vulgar words will result in a competitor's suspension from the current competition or a delegate's suspension from a delegate meeting.

Forbidden words are racial, ethnic, gender, or sexuality-based slurs. If a word does not fall into the above categories but is regularly used as a slur, it is a slur. Usage of a slur will result in the user's institution being temporarily suspended for three months (see). Usage of a slur will also result in the permanent suspension of the user.

Put-Downs and Insults

If it is found that a competitor is using verbal insults to degrade another team or another competitor, that competitor will be given a warning and asked to stop. The competitor's coach will also be notified when the warning is given to the competitor.

If the competitor continues to verbally insult or degrade other teams and competitors, the competitor will be asked to leave the competition.

Respect to Delegates

Delegates have spent their own time building and managing competitions, so they deserve the respect of the competitors. When a delegate is speaking to a room full of competitors, it is expected that the competitors are listening. This means that competitors close their laptops, are looking at the delegate, and are actively engaged in the conversation.

If a competitor is found to be rude or verbally insulting to a delegate, the "Put-Downs and Insults" rule is used. The competitor will be warned once (and their coach notified), and if the verbal harassment continues, the competitor will be asked to leave the competition.

Respect to Coaches

Coaches spend their time managing their teams and corralling competitors from their school, and so they deserve the upmost respect of the competitors. Competitors, even those on separate teams from a coach, should be kind and polite to said coach.

If a competitor is found to be rude or verbally insulting to a coach, the "Put-Downs and Insults" rule is used. The competitor will be warned once (and their coach notified), and if the verbal harassment continues, the competitor will be asked to leave the competition.

Prohibited Substances

The Fairfield Programming Association uses schools as the primary location for hosting events and competitions. This means that the FPA must follow many of the rules and guidelines put in place by each of these institutions, especially when it comes to controlled substances.

Alcohol Products

Alcohol and other illicit drinks are strictly prohibited from competitions and events. It is also expected that competitors, delegates, coaches, and other members of an organization be sober during the events. Failure to comply with either of these rules will result in the individual being asked to leave the event, and their organization may be subject to disciplinary action.

Marijuana and Narcotic Items

Tobacco and Nicotine Products

Usage of Tobacco and Nicotine products is strictly prohibited during competition, delegate meetings, and other FPA events. If a competitor, delegate, coach, or other member of an organization is found to be using one of these products, they may be asked to leave the event and their organization may be subject to disciplinary action.

Electronic vapes, hookahs, cigarettes, and cigars all fall under the category of tobacco and nicotine products. Devices that are not listed here but act in a similar fashion to the above products can be categorized as such; this is under the discretion of the delegates.

The Ferragu Clause

During the annual banquet, some prohibited substances are allowed. These substances include cigars, champagne, and liqueurs such as whiskey, brandy, baileys, etc. These are only to be used in a celebratory manner and should not be used to intentionally cause explicit behavior. The choice to consume or use these substances is at the discretion of the individual's parents if they are under 18 years of age or under their own discretion if the individual is past the legal age.

Institution Suspension

Teams and institutions can be suspended for a number of reasons. But, all of these reasons share the same idea, the institution was not following the rules and the general ideas that the Fairfield Programming Association supports.

Competition Suspension

Suspension from a competition is the most basic punishment that a team can earn

Three Month Suspension

One Year Suspension

Two Year Suspension

Delegate Rules

Delegates are meant to represent schools, but they are not meant to represent the schools interests. Rather, delegates represent the Fairfield Programming Association and their actions should be in the best interest of the association.

Number of Delegates

Each school or organization is allowed to send two delegates at most. Alternatively, schools are allowed to send one delegate, and this delegate will have the power of two votes. If a school does not send any delegates, they will not have the ability to vote during the meeting.

Meeting Times

Meetings will be organized based on who will be able to attend. If there is at least a $\frac{2}{3}$ majority of delegates that are able to attend, a meeting will be held. If less than $\frac{2}{3}$ of the delegates are able to attend, meeting and voting will not be allowed.

Chief Delegate

At the start of every meeting, there will be a chief delegate that needs to be elected. For the initial meeting of every year, this will be the first delegate of Greens Farms Academy. This is done to prevent chaotic voting because the relations of the delegates may be uncertain at the beginning of the year.

Chief Delegate Responsibilities

The job of the chief delegate is to keep the meetings on track and stop the delegates from promoting inappropriate behavior. To do this, the chief delegate will track how long each of the delegates have spoken for and will also call votes whenever there needs to be one.

Chief Delegate Voting

The chief delegate is unable to vote in any of the decisions and they are not allowed to voice their opinions. They are there to act as a mitigator for chaos and a leader to keep the meetings on track.

Speaking Times

At the start of a meeting, delegates will be able to write their name on a docket. This docket will list the order of speaking for each of the delegates. Delegates are only allowed to speak for two minutes each, but are allowed to write their names back on the docket once their speaking has finished. The docket acts as a queue, so each new addition to the docket will occur at the bottom and the first speakers will be on the top.

Contest Rules

Contestants can use the python documentation but cannot use question and answer sites like stack overflow.

<https://www.python.org/doc/>

Problem Rules

Points and Scoring

The scoring behind questions will be split into two different categories. There will be some questions that scoring will be based off of time-spent. This means that the question will target a certain amount of time that it will take for a user to compete, and the score will be based on this amount of time. The other option for scoring is difficulty based. This will be mostly for problems where solutions are non-obvious, and target a more niche section of competitors that may know about a certain topic.

Time Spent Scoring

The time spent scoring system works on an exponential scale. This means that the amount of points that a problem is worth is directly impacted to how long a problem should take, and it is modeled by the below function. The output of this function will be rounded to the tens-place and the resulting number will be the number of points the question is worth.

$$p = 200 \times 1.1^t - 200$$

0	1	2	3	4	5	6	7	8	9	10
0	20	40	70	90	120	150	190	230	270	320

Difficulty Based Scoring

Difficulty scoring is more subjective. It combines the time spent scoring with a bit of personal influence. For each difficult concept that is covered in the problem, the score will be increased by 110% and the resulting score, after all of the multipliers, will be rounded to the tens-place and the resulting number will be the point amount for the problem.

An example of calculating the point value of a five minute problem with two difficult concepts is shown below. The difficult concepts that the problem contains could be anything from text processing to high-level math to advanced binary calculations.

$$\sim 120 = 200 \times 1.1^5 - 200$$

$$\sim 150 = 120 \times 110\% \times 110\%$$

So the final count would be 150 points for the five minute problem with two difficult concepts.

Style Guide

Each problem will have a description split into multiple sections. The first section of the description will always be the introduction. This section will introduce the problem and describe the basic ideas behind it. The second section of the description will always be the “Input and Output” section. This section will describe how the input that the solution takes in is formatted, and how the output that the solution produces should be formatted. Finally, there can optionally be a section describing some background information about the question such as the history or use of the problem in the real world.

Problem Input and Output

Each problem will have the ability to take in input, and all problems will produce output. The input and output of each problem will be described in a section of the problem’s description entitled, “Input and Output”.

Problem Accessibility

Each problem will be accessible to anyone who is currently enrolled in the College Board’s AP Computer Science and has completed the first 6 weeks of the curriculum. For individuals not enrolled in this course, the problems will require a solid knowledge of computer science and a rudimentary understanding of python programming.

Solution Formatting

The solutions to each of the problems will be formatted using the [python style guide](#). These solutions will be thoroughly tested and will contain detailed comments about what each section of the program is doing. The variable names will always use underscores with all lowercase characters rather than camelcase naming.

Solution Design

Every problem will be designed for python 3. This means that problems will not rely on types, object oriented programming, or any other programming topics that python 3 was not explicitly designed for. Every solution to the problem will be written in python 3 as well.

Solution Input and Output

The input and output of the solution program will be static and functional. This means that given a certain input (for instance ‘6’), the program will always output the same thing. In other words, random generation, time-based functions, or any methods based on external information are not permitted. This is mainly for solution testing purposes, but it is a requirement for all solutions and problems.

Libraries and Imports

Problems will be designed to be solved without the use of libraries. The only library permitted for use in the solutions to problems is the python math library. The only functions in this library that permitted for use are listed below in the allowed methods section.

Problem Length

Problem solutions will be no longer than 80 lines.

Allowed Methods in the Standard Library

The problems will be designed to only use the following methods from the python 3 standard library and the methods in the below math library. If a problem can only be solved using a method not on this list, it should be reported by a competitor.

abs()	float()	oct()
aiter()	format()	open()
all()	frozenset()	ord()
any()	getattr()	pow()
anext()	globals()	print()
ascii()	hasattr()	property()
bin()	hash()	range()
bool()	help()	repr()
breakpoint()	hex()	reversed()
bytearray()	id()	round()
bytes()	input()	set()
callable()	int()	setattr()
chr()	isinstance()	slice()
classmethod()	issubclass()	sorted()
compile()	iter()	staticmethod()
complex()	len()	str()
delattr()	list()	sum()
dict()	locals()	super()
dir()	map()	tuple()
divmod()	max()	type()
enumerate()	memoryview()	vars()
eval()	min()	zip()
exec()	next()	__import__()
filter()	object()	

Allowed Methods in the Math Library

The problems will be designed to only use the following methods from the python 3 math library and the methods in the above standard library. If a problem can only be solved using a method not on this list, it should be reported by a competitor.

<code>math.acos()</code>	<code>math.fabs()</code>	<code>math.log10()</code>
<code>math.acosh()</code>	<code>math.factorial()</code>	<code>math.log1p()</code>
<code>math.asin()</code>	<code>math.floor()</code>	<code>math.log2()</code>
<code>math.asinh()</code>	<code>math.fmod()</code>	<code>math.modf()</code>
<code>math.atan()</code>	<code>math.frexp()</code>	<code>math.nextafter()</code>
<code>math.atan2()</code>	<code>math.fsum()</code>	<code>math.perm()</code>
<code>math.atanh()</code>	<code>math.gamma()</code>	<code>math.pow()</code>
<code>math.ceil()</code>	<code>math.gcd()</code>	<code>math.prod()</code>
<code>math.comb()</code>	<code>math.hypot()</code>	<code>math.radians()</code>
<code>math.copysign()</code>	<code>math.isclose()</code>	<code>math.remainder()</code>
<code>math.cos()</code>	<code>math.isinf()</code>	<code>math.sin()</code>
<code>math.cosh()</code>	<code>math.isinfinite()</code>	<code>math.sinh()</code>
<code>math.degrees()</code>	<code>math.isnan()</code>	<code>math.sqrt()</code>
<code>math.dist()</code>	<code>math.isqrt()</code>	<code>math.tan()</code>
<code>math.erf()</code>	<code>math.lcm()</code>	<code>math.tanh()</code>
<code>math.erfc()</code>	<code>math.ldexp()</code>	<code>math.trunc()</code>
<code>math.exp()</code>	<code>math.lgamma()</code>	<code>math.ulp()</code>
<code>math.expm1()</code>	<code>math.log()</code>	

Example Problems

Example 1 - Music Mixup

Intro

For synthesizers to work, they need to know what notes have certain values. Throughout the years, people have developed a string-notation system for these notes, and it is your job to calculate the note's integer value when given the note using the string-notation. To find the integer value of a note, you first need to understand how notes are stored.

C	=	0
C#/D-	=	1
D	=	2
D#/E-	=	3
E/F-	=	4
E#/F	=	5
F#/G	=	7
G#/A-	=	8
A	=	9
A#/B-	=	10
B	=	11
B#	=	12

Notes are stored using letters, accidental symbols, and numbers denoting the octave. The first character is always the letter, denoting the core value of the note. This letter can be anything A-G, but will always be uppercase. After the letter will be the accidental, which can be either a '#' or a '-'. This accidental is optional, and so not all string-notation notes will contain one. The final character of a string-notation note will always be the octave. An octave character can be any number ranging from 0-9.

In the case of this question, you do not have to think about the octave, only the note value inside of the octave. Your program should return the following values from above, but should do this in a programmatic way- no maps.

Input and Output Testing

The program should be able to take in the following values and output the following outputs.

Input = Output

A#3	=	10
A-3	=	8
B3	=	11
B#3	=	12
C2	=	0
D3	=	2
E5	=	4
F7	=	5
G2	=	7

Sample Solution

```
def GetNote(input):  
  
    # Make Sure Note Exists  
    if (len(input) < 2):  
        return -1  
  
    if (len(input) > 3):  
        return -1  
  
    # Dissect the String to Char  
    noteChar = ord(input[0])  
  
    # Find the Note Value. { A: 9, B: 11, C: 0, D: 2, E: 4, F: 5, G: 7}  
    finalNotes = [ 9, 11, 0, 2, 4, 5, 7 ]  
    finalNoteValue = finalNotes[noteChar - 65]  
  
    # Check if note has accidental  
    if (len(input) == 3):  
  
        # Dissect the String  
        accidentalChar = input[1]  
  
        # If it has an accidental, apply accidental  
        if (accidentalChar == '#'):  
            finalNoteValue += 1  
  
        if (accidentalChar == '-'):  
            finalNoteValue -= 1  
  
    # Return the Final Value  
    return finalNoteValue
```


Example 2

Intro

Here's a question to see how well people know how to use loops and conditionals, "Can you write a program that prints all the numbers from 1 to 100?". But there is a catch, every multiple of three, print "Fizz", instead of the number. And for every multiple of five, print out "Buzz" instead of the number. Finally, if the number is a multiple of both three and five, print out "FizzBuzz". Each number should be separated by a newline, and each word should also be separated by a newline. An example of this can be shown in the figure below.

Input and Output Testing

Since there is no special input for this question, the only output that should be verified is below.

1	Fizz	71
2	37	Fizz
Fizz	38	73
4	Fizz	74
Buzz	Buzz	FizzBuzz
Fizz	41	76
7	Fizz	77
8	43	Fizz
Fizz	44	79
Buzz	FizzBuzz	Buzz
11	46	Fizz
Fizz	47	82
13	Fizz	83
14	49	Fizz
FizzBuzz	Buzz	Buzz
16	Fizz	86
17	52	Fizz
Fizz	53	88
19	Fizz	89
Buzz	Buzz	FizzBuzz
Fizz	56	91
22	Fizz	92
23	58	Fizz
Fizz	59	94
Buzz	FizzBuzz	Buzz
26	61	Fizz
Fizz	62	97
28	Fizz	98
29	64	Fizz
FizzBuzz	Buzz	Buzz
31	Fizz	
32	67	
Fizz	68	
34	Fizz	
Buzz	Buzz	

Sample Solution

```
for fizzbuzz in range(100):  
  
    if fizzbuzz % 3 == 0 and fizzbuzz % 5 == 0:  
        print("fizzbuzz")  
        continue  
  
    elif fizzbuzz % 3 == 0:  
        print("fizz")  
        continue  
  
    elif fizzbuzz % 5 == 0:  
        print("buzz")  
        continue  
  
    print(fizzbuzz)
```