#	Old Code	#	New Code
1	`# Keywords & syntax demo (A)`	1	`# Keywords & syntax demo (B)`
1		1	
2	`import math`	2	`import math`
3	`from math import pi as circle_pi`	3	`from math import pi as circle_pi`
4		4	
5	`class Example:`	5	`class Example:`
6	`    def __init__(self, value: int = 0) -> None:`	6	`    def __init__(self, value: int = 1) -> None:  # Changed default from 0 to 1`
6	`        self.value = value`	6	`        self.value = value`
7		7	
8	`    def compute(self) -> float:`	8	`    def compute(self) -> float:`
9	`        if self.value > 0:`	9	`        if self.value >= 0:  # Changed > to >=`
9	`            for i in range(1, 10):`	9	`            for i in range(1, 10):`
10	`                while i < 5:`	10	`                while i < 6:  # Changed 5 to 6`
10	`                    try:`	10	`                    try:`
11	`                        assert i != 3, "Unlucky number"`	11	`                        assert i != 4, "Unlucky number"  # Changed 3 to 4`
11	`                        yield i`	11	`                        yield i`
12	`                        break`	12	`                        break`
13	`                    except AssertionError as e:`	13	`                    except AssertionError as e:`
14	`print(f"Caught: {e}")`	14	`print(f"Error: {e}")  # Changed message`
14	`                        continue`	14	`                        continue`
15	`                    finally:`	15	`                    finally:`
16	`                        pass`	16	`                        pass`
17	`        elif self.value == 0:`	17	`        elif self.value == -1:  # Changed 0 to -1`
17	`            return None`	17	`            return None`
18	`        else:`	18	`        else:`
19	`            raise ValueError("Negative!")`	19	` raise ValueError("Too negative!") # Changed error message and a liot of other things and many more sthings and erhlghs eskjrhg ewg ewkh lk4w5ypow45klthq3 k45hkjlw hkj54wnt 3q5t 5iuyg4wiu hq5k4nt kjl35wht jgwhj we hjl ghwergewrjgh erwjgh ewrkjgh erwkljgh ewrrg ewrgj herwg erwrgh ewjrgh ewrjgh werjlgjj hwerkljgh wergj hewrkjgh wergh wergh ewrkjlgherw gerwkjlhgkl wergkjlw ewrjgh wergh werkjgh kj jkerhgj wegr`
19		19	
20	`def main():`	20	`def main():`
21	`    e = Example(2)`	21	`    if a:`

22	result = [x for x in e.compute() if x % 2 == 0]	None	
23	print("Results:", result)	None	
21	match e.value:	21	match e.value:
22	case 0:	22	case 0:
23	print("Zero")	23	print("Zero")
24	def inner(*args, **kwargs):	24	def inner(*args, **kwargs):
25	global x	25	global x
26	nonlocal result	26	nonlocal result
27	x = lambda y: y ** 2	27	x = lambda y: y + 1 # Changed expression
28	print({k: v for k, v in kwargs.items()})	28	print({k: v.upper() for k, v in kwargs.items()}) # Added .upper()
27	return x(args[0]) if args else None	27	return x(args[0]) if args else None
28		28	
29	print(inner(4, key='val'))	29	print(inner(3, key='val')) # Changed arg
29		29	
30	if __name__ == "__main__":	30	if __name__ == "__main__":
31	main()	31	main()
32		32	
33		33	
34	old_part = sanitize(old_line[i1:i2])	34	old_part = "".join(sanitize(tok) for tok in old_tokens[i1:i2])
35	new_part = sanitize(new_line[j1:j2])	35	new_part = "".join(sanitize(tok) for tok in new_tokens[j1:j2])
34		34	
35	\ No newline at end of file	35	\ No newline at end of file