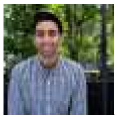
gh faizywaizy CPS_847_Assignment2 Docs Support Blog



▼ Showing 3 of 4 files from the diff.

test.py created.	Newly tracked file	100.00%	< 100.00% >	ø
operations.py created.	Newly tracked file	100.00%	< 100.00% >	ø
main.py created.	Newly tracked file	100.00%	< 100.00% >	ø

▶ Other files ignored by Codecov

test.py	17	100.00%	< 100.00% >	ø

```
@@ -0,0 +1,17 @@
+   import unittest
+   import main
+   import operations
+
+   class Test(unittest.TestCase):
+       def setUp(self):
+           pass
+       def test1(self):
+           testNumbers = [6, 5, 0.32, 0.82, 0, 65, 0.5, 32, 8, 0.506, 432, 456.2, 0]
+           main.testCode(testNumbers)
+       def test2(self):
+           self.assertEqual(operations.multiply(9,2), 18)
+       def test3(self):
+           self.assertEqual(operations.divide(18,2), 9)
+
+   if __name__ == '__main__':
+       unittest.main()
```

operations.py	19	100.00%	< 100.00% >	ø

```
@@ -0,0 +1,19 @@
+   def halved(value):
+       return value/2
+
+   def quartered(value):
```

```
+        return value/4
+
+    def divide(value,divisor):
+        if divisor == 0:
+            raise ZeroDivisionError('Cannot divide by zero.')
+        return  (value/divisor)
+
+    def squared(value):
+        return value*value
+
+    def cubed(value):
+        return value*value*value
+
+    def multiply(value,multiplier):
+        return value*multiplier
```

main.py	18	100.00%	< 100.00% >	ø

```
@@ -0,0 +1,18 @@
+    import operations
+    testNumbers = [6, 5, 0.32, 0.82, 0, 65, 0.5, 32, 8, 0.506, 432, 456.2, 0]
+
+    def testCode(data):
+        for number in data:
+            if number == 0:
+                print("Skipping 0\n")
+                continue
+            try:
+                print("{0} squared is {1}\n".format(number,operations.squared(number)))
+                print("{0} cubed is {1}\n".format(number,operations.cubed(number)))
+                print("{0} multiplied by 3 is {1}\n".format(number,operations.multiply(nu
+                print("{0} halved is {1}\n".format(number,operations.halved(number)))
+                print("{0} quartered is {1}\n".format(number,operations.quartered(number)
+                print("{0} divided by zero is {1}\n".format(number,operations.divide(numb
+
+            except ZeroDivisionError:
+                print("Divisor cannot be zero\n")
```