Step 1

Convert from miles to kilometers.

Conversion generally is ((X + k1) * C) + k2

In our case k1 and k2 are 0. So we just get X * C

Demo - lookup conversion from miles to kilometers

```

    # Step 1 - constants

    miles = 3
    conv = 1.60934
    km = miles * conv

```

Demo - of this as a visualization

Step 2 - Input with error

```

    # Step 2 - will error with type error

    print ( "Enter Miles" )

    miles = input()

    conv = 1.60934
    km = miles * conv

    print ( "km = {}".format(km) )

```

Step 3 - Fixed error / Types

```

    # Step 3 - inline after fixing type

```

```
4    print ( "Enter Miles" )
5
6    miles_str = input()
7    miles = int(miles_str)
8    conv = 1.60934
9    km = miles * conv
10
11   print ( "km = {}".format(km) )
```

Step 4 - Make a function

```
1
2    # Step 4 - After making a function
3
4    def mi_to_km ( mi ):
5        conv = 1.60934
6        km = mi * conv
7        return (km)
8
9    print ( "Enter Miles" )
10
11   miles_str = input()
12   miles = int(miles_str)
13
14   km = mi_to_km(miles)
15
16   print ( "km = {}".format(km) )
```

Step 5 - Make Reusable Code

step-5.py:

```
1
2    # Step 5 - with function and a test.
3
4    import mi_to_km
5
6    print ( "Enter Miles" )
7
8    miles_str = input()
9    miles = int(miles_str)
10
11   km = mi_to_km.mi_to_km(miles)
```

```
12
13    print ( "km = {}".format(km) )
```

conv/mi_to_km.py:

```
 1
 2    # mi_to_km converts from miles as an integer or float to kilometers.
 3    def mi_to_km ( mi ):
 4        conv = 1.60934
 5        km = mi * conv
 6        return (km)
 7
 8    # Automated Test
 9    if __name__ == "__main__":
10        n_err = 0
11        x = mi_to_km ( 3 )
12        if x !=  4.82802:
13            n_err = n_err + 1
14            print ( "Error: Test 1: conversion not working, expected {} got {}".f
15        x = mi_to_km ( 0 )
16        if x != 0:
17            n_err = n_err + 1
18            print ( "Error: Test 2: conversion not working, expected {} got {}".f
19
20        if n_err == 0 :
21            print ( "PASS" )
22        else:
23            print ( "FAILED" )
24
```

Step 6 - Add documentation

This is really a little step in this program - but a really important one for this class..

```
 1
 2    # Author: Philip Schlump
 3    # Email: pschlump@uwyo.edu
 4
 5    # Main program to read in values and convert from miles (mi) to kilometers (k
 6
 7    # Step 5 - with function and a test.
 8
 9    import mi_to_km
10
```

```
11    print ( "Enter Miles" )
12
13    miles_str = input()
14    miles = int(miles_str)
15
16    km = mi_to_km.mi_to_km(miles)
17
18    print ( "km = {}".format(km) )
19
```