Appendix 1: TDD

Making Tests Pass

- Every time you add a new test case, you have to make it pass by making the code more general.

Add test

```
public void testTwo() throws Exception {
    int factors[] = PrimeFactorizer.factor(2);
    assertEquals(1, factors.length);
    assertEquals(2, factors[0]);
}
```

Compiles
test fails

```
public class PrimeFactorizer {
    public static int[] factor(int multiple) {
        return new int[0];
    }
}
```

Pass
simplest
change

```
public static int[] factor(int multiple) {
    return new int[] {2};
}
```

http://www.objectmentor.com/resources/articles/craftsman5.pdf

Add test

```
public void testThree() throws Exception {
    int factors[] = PrimeFactorizer.factor(3);
    assertEquals(1, factors.length);
    assertEquals(3, factors[0]);
}
```

Pass
but wrong!

```
public static int[] factor(int multiple) {
    if (multiple == 2) return new int[] {2};
    else return new int[] {3};
}
```

Pass
and right

```
public static int[] factor(int multiple) {
    return new int[] {multiple};
}
```

http://www.objectmentor.com/resources/articles/craftsman5.pdf

Add test

```
public void testFour() throws Exception {
    int factors[] = PrimeFactorizer.factor(4);
    assertEquals(2, factors.length);
    assertEquals(2, factors[0]);
    assertEquals(2, factors[1]);
}
```

http://www.objectmentor.com/resources/articles/craftsman5.pdf

Pass
but dirty

```
public class PrimeFactorizer {
    public static int[] factor(int multiple) {
        int currentFactor = 0;
        int factorRegister[] = new int[2];
        for (; (multiple % 2) == 0; multiple /= 2)
            factorRegister[currentFactor++] = 2;
        if (multiple != 1)
            factorRegister[currentFactor++] = multiple;
        int factors[] = new int[currentFactor];
        for (int i = 0; i < currentFactor; i++)
            factors[i] = factorRegister[i];
        return factors;
    }
}
```

http://www.objectmentor.com/resources/articles/craftsman5.pdf

Refactor
until clean

```
public class PrimeFactorizer {
    private static int factorIndex;
    private static int[] factorRegister;

    public static int[] factor(int multiple) {
        initialize();
        findPrimeFactors(multiple);
        return copyToResult();
    }

    private static void initialize() {
        factorIndex = 0;
        factorRegister = new int[2];
    }

    private static void findPrimeFactors(int multiple) {
        for (; (multiple % 2) == 0; multiple /= 2)
            factorRegister[factorIndex++] = 2;
        if (multiple != 1)
            factorRegister[factorIndex++] = multiple;
    }

    private static int[] copyToResult() {
        int factors[] = new int[factorIndex];
        for (int i = 0; i < factorIndex; i++)
            factors[i] = factorRegister[i];
        return factors;
    }
}
```

http://www.objectmentor.com/resources/articles/craftsman5.pdf

Course Material

- http://www.objectmentor.com/resources/articles/craftsman5.pdf