PAL Worksheet Method overloading CSC 15

NOTE: This worksheet should only be done after "Standard Library".

It is common to write methods with the same name but with different parameters. An example is the Math static utility function Math.abs. There are four versions in the Math class:

```
public static double abs(double a)
public static float abs(float a)
public static int abs(int a)
public static long abs(long a)
```

When a user calls Math.abs(x), Java automatically invokes the one that matches x's type.

1) Write a class called BaseConversion. It should contain public static methods called toInt, toHex, and toBinary. There should be no main method in the class. This class is simply a collection of utility methods.

   The method toInt should return an int while toHex and toBinary should each return a String. There should be two versions of toHex: one that takes a String and one that takes an int. There should be two versions of toBinary: one that takes a String and one that takes an int. There should be only one version of toInt, taking a String.

   toInt(s) should interpret s as a hex string and return its int value.
   toHex(s) should interpret s as a binary string and return a hex string.
   toHex(x) should return a hex String with the same value as int x.
   toBinary(s) should interpret s as a hex string and return a binary string.
   toBinary(x) should return a binary String with the same value as int x.

   You should use the Integer utility methods to do most of the work.

2) Write a separate program to test that your utility class works correctly. Make sure the file BaseConversion.java is in the same folder/directory as your program, otherwise it won't be found.

3) If time remains, either alone or in pairs, go to https://codestepbystep.com. Follow Practice! > Java > parameters and return. Do the any of the ParameterMystery exercises. Be sure to deskcheck as you go by writing down what the current values are inside the current scope.