Project 5

Project 5 was relatively easy, which was surprising as classes is a brand-new concept for me. By following the diagrams on the spec, I was able to declare private and public methods in their respective header files, and use the correct syntax to call them in the cpp files for most of them. Initially, I forgot to write a default constructor, so instead I initialized the first method like so ZodiacReader::ZodiacReader(Date date):mDate(date). Once this worked. I was able to use the getMonth() and getDay() with mDate from the Date.cpp and Date.h. By using a series of if statements, I programmed the code to return the sign corresponding to these dates. Then, to determine if the date was on the cusp, I first needed to define what it means for a date to be on the cusp. This is where I had most trouble, but I figured it out by drawing a diagram. At first, I only considered a total of 3 dates per month that were considered to be on the cusp. This was faulty logic, as I did not consider cusp dates that overlap with 2 adjacent signs. Therefore, there were actually 4 total dates per month.

Once I figured this out, to return cusp sign proved to be tricky. For example, if born on April 18th, the corresponding sign is Aries, but is on the cusp of Taurus. However, if you were born near the start of the Aries sign, say March 21, the sign is Aries, but the cusp is now the sign that comes previously, in this case, Pisces. To implement this into my code, I first checked if the date was on the cusp. Then, if it was one sign, and was in the starting month (in the proposed case, March), it returns the previous sign in the cycle (in the proposed case, Pisces). If the date was in the ending month (in the proposed case, April), then it returns the next sign in the cycle (in the proposed case, Taurus).

Finally, for the stringify method, I encountered a bug with “stringifying” the cuspSign. It worked just fine for checkSign, so I figured something was wrong with cuspSign code. There wasn’t. Instead, I realized that in the method I created sign of type ZodiacReader::Sign. Therefore, I could use sign==ZodiacReader::Sign, instead of ZodiacReader::checkSign()==Sign and ZodiacReader::cuspSign()==Sign. By doing this, I could account for all signs returned, rather than considering them as separate methods.

For the test cases, I wrote over a 100 or so. I tested all cusp dates, and general dates that should return the sign of the interval they fall in. I’ve indicated the ones I tested for cusp and also the ones that check if the beginning and end of the months return the correct sign. The ones without comments are general tests and will either have an object called genTest followed by an integer, and/or no comments.

*Here are the Test Cases:*

// test code

    //////////////////////////////////////////////////

    //      ARIES             MARCH 21 - APRIL 19  ///

    //////////////////////////////////////////////////

    Date mar21( 3, 21 );

    ZodiacReader read( mar21 );

    assert( read.stringifySign( read.checkSign() ) == "Aries" );            //Sign: Aries

    assert( read.checkSign( ) == ZodiacReader::ARIES );                     //One exact start day

    assert( read.onCusp( ) );                                               //Cusp: Pisces

    assert( read.cuspSign()==ZodiacReader::PISCES);

    assert( read.stringifySign( read.cuspSign() ) == "Pisces" );

    Date mar22( 3, 22 );

    ZodiacReader read1( mar22 );

    assert( read1.stringifySign( read1.checkSign() ) == "Aries" );      //Sign: Aries

    assert( read1.checkSign( ) == ZodiacReader::ARIES );                //One dat after start day

    assert( read1.onCusp( ) );                                          //Cusp: Pisces

    assert( read1.cuspSign()==ZodiacReader::PISCES);

    assert( read1.stringifySign( read1.cuspSign() ) == "Pisces" );

    Date mar23( 3, 23 );

    ZodiacReader read2( mar23 );

    assert( read2.stringifySign( read2.checkSign() ) == "Aries" );

    assert( read2.checkSign( ) == ZodiacReader::ARIES );

    assert( !read2.onCusp( ) );

assert( read2.cuspSign()==ZodiacReader::ARIES); //if not on cusp, return ARIES by default

    assert( read2.stringifySign( read2.cuspSign() ) == "Aries" );

    Date mar30( 3, 30 );

    ZodiacReader read3( mar30 );

    assert( read3.stringifySign( read3.checkSign() ) == "Aries" );

    assert( read3.checkSign( ) == ZodiacReader::ARIES );

    assert( !read3.onCusp( ) );

    Date mar31( 3, 31 );

    ZodiacReader read4( mar31 );                                    //End of month returns correct sign

    assert( read4.stringifySign( read4.checkSign() ) == "Aries" );

    assert( read4.checkSign( ) == ZodiacReader::ARIES );

    assert( !read4.onCusp( ) );

    Date april1( 4, 1 );

    ZodiacReader read5( april1 );                                   //Beginning of month returns correct sign

    assert( read5.stringifySign( read5.checkSign() ) == "Aries" );

    assert( read5.checkSign( ) == ZodiacReader::ARIES );

    assert( !read5.onCusp( ) );

    Date april12( 4, 12 );

    ZodiacReader read6( april12 );

    assert( read6.stringifySign( read6.checkSign() ) == "Aries" );

    assert( read6.checkSign( ) == ZodiacReader::ARIES );

    assert( !read6.onCusp( ) );

    Date april18( 4, 18 );                                          //Sign: Aries

    ZodiacReader read7( april18 );                                  //One day before end date

    assert( read7.stringifySign( read7.checkSign() ) == "Aries" );  //Cusp: Taurus

    assert( read7.checkSign( ) == ZodiacReader::ARIES );

    assert( read7.onCusp( ) );

    assert( read7.cuspSign()==ZodiacReader::TAURUS);

    assert( read7.stringifySign( read7.cuspSign() ) == "Taurus" );

    Date april19( 4, 19 );                                          //Sign: Aries

    ZodiacReader read8( april19 );                                  //On exact end date for Aries, one day before exact start date for Taurus

    assert( read8.stringifySign( read8.checkSign() ) == "Aries" );  //Cusp: Taurus

    assert( read8.checkSign( ) == ZodiacReader::ARIES );

    assert( read8.onCusp( ) );

    assert( read8.cuspSign()==ZodiacReader::TAURUS);

    assert( read8.stringifySign( read8.cuspSign() ) == "Taurus" );

    //////////////////////////////////////////////////

    //      TAURUS             APRIL 20 - MAY 20   ///

    //////////////////////////////////////////////////

    Date april20( 4, 20 );

    ZodiacReader read9( april20 );                                  //Sign: Taurus

    assert( read9.stringifySign( read9.checkSign() ) == "Taurus" ); //On exact start date for Taurus, on one day after end date for Aries

    assert( read9.checkSign( ) == ZodiacReader::TAURUS );           //Cusp: Aries

    assert( read9.onCusp( ) );

    assert( read9.cuspSign( ) == ZodiacReader::ARIES );

    assert( read9.stringifySign( read9.cuspSign() ) == "Aries" );

    Date april21( 4, 21 );

    ZodiacReader read10( april21 );                                     //Sign: Taurus

    assert( read10.stringifySign( read10.checkSign() ) == "Taurus" );   //One day after exact start date

    assert( read10.checkSign( ) == ZodiacReader::TAURUS );              //Cusp: Aries

    assert( read10.onCusp( ) );

    assert( read10.cuspSign( ) == ZodiacReader::ARIES );

    assert( read10.stringifySign( read10.cuspSign() ) == "Aries" );

    Date april26( 4, 26 );                                              //My birthday! :)

    ZodiacReader genTest7( april26 );

    assert( genTest7.stringifySign( genTest7.checkSign() ) == "Taurus" );

    assert( genTest7.checkSign( ) == ZodiacReader::TAURUS );

    assert( !genTest7.onCusp( ) );

    Date april30( 4, 30 );                                              //End of month returns correct sign

    ZodiacReader read11( april30 );

    assert( read11.stringifySign( read11.checkSign() ) == "Taurus" );

    assert( read11.checkSign( ) == ZodiacReader::TAURUS );

    assert( !read11.onCusp( ) );

    Date may1( 5, 1 );                                                  //Beginning of month return correct sign

    ZodiacReader read12( may1 );

    assert( read12.stringifySign( read12.checkSign() ) == "Taurus" );

    assert( read12.checkSign( ) == ZodiacReader::TAURUS );

    assert( !read12.onCusp( ) );

    Date may17( 5, 17 );

    ZodiacReader genTest8( may17 );

    assert( genTest8.stringifySign( genTest8.checkSign() ) == "Taurus" );

    assert( genTest8.checkSign( ) == ZodiacReader::TAURUS );

    assert( !genTest8.onCusp( ) );

    Date may19( 5, 19 );                                                //Sign: Taurus

    ZodiacReader read13( may19 );                                       //One day before exact end date

    assert( read13.stringifySign( read13.checkSign() ) == "Taurus" );   //Cusp: Gemini

    assert( read13.checkSign( ) == ZodiacReader::TAURUS );

    assert( read13.onCusp( ) );

    assert( read13.cuspSign( ) == ZodiacReader::GEMINI );

    assert( read13.stringifySign( read13.cuspSign() ) == "Gemini" );

    Date may20( 5, 20 );                                                //Sign: Taurus

    ZodiacReader read14( may20 );                                       //On exact end date for Taurus, one day before start date for Gemini

    assert( read14.stringifySign( read14.checkSign() ) == "Taurus" );   //Cusp: Gemini

    assert( read14.checkSign( ) == ZodiacReader::TAURUS );

    assert( read14.onCusp( ) );

    assert( read14.cuspSign( ) == ZodiacReader::GEMINI );

    assert( read14.stringifySign( read14.cuspSign() ) == "Gemini" );

    //////////////////////////////////////////////////

    //      GEMINI             MAY 21 - JUNE 20     //

    //////////////////////////////////////////////////

    Date may21( 5, 21 );                                                //Sign: Gemini

    ZodiacReader read15( may21 );                                       //On exact start date for Gemini, one day after exact end date for Taurus

    assert( read15.stringifySign( read15.checkSign() ) == "Gemini" );   //Cusp: Taurus

    assert( read15.checkSign( ) == ZodiacReader::GEMINI );

    assert( read15.onCusp( ) );

    assert( read15.cuspSign( ) == ZodiacReader::TAURUS );

    assert( read15.stringifySign( read15.cuspSign() ) == "Taurus" );

    Date may22( 5, 22 );

    ZodiacReader read16( may22 );                                       //Sign: Gemini

    assert( read16.stringifySign( read16.checkSign() ) == "Gemini" );   //One day after exact start date for Gemini

    assert( read16.checkSign( ) == ZodiacReader::GEMINI );              //Cusp: Taurus

    assert( read16.onCusp( ) );

    assert( read16.cuspSign( ) == ZodiacReader::TAURUS );

    assert( read16.stringifySign( read16.cuspSign() ) == "Taurus" );

    Date may27( 5, 27 );

    ZodiacReader genTest9( may27 );

    assert( genTest9.stringifySign( genTest9.checkSign() ) == "Gemini" );

    assert( genTest9.checkSign( ) == ZodiacReader::GEMINI );

    assert( !genTest9.onCusp( ) );

    Date may31( 5, 31 );                                                //End of month returns correct sign

    ZodiacReader read17( may31 );

    assert( read17.stringifySign( read17.checkSign() ) == "Gemini" );

    assert( read17.checkSign( ) == ZodiacReader::GEMINI );

    assert( !read17.onCusp( ) );

    Date june1( 6, 1 );                                                 //Beginning of month returns correct sign

    ZodiacReader read18( june1 );

    assert( read18.stringifySign( read18.checkSign() ) == "Gemini" );

    assert( read18.checkSign( ) == ZodiacReader::GEMINI );

    assert( !read18.onCusp( ) );

    Date june12( 6, 12 );

    ZodiacReader genTest10( june12 );

    assert( genTest10.stringifySign( genTest10.checkSign() ) == "Gemini" );

    assert( genTest10.checkSign( ) == ZodiacReader::GEMINI );

    assert( !genTest10.onCusp( ) );

    Date june19( 6, 19 );                                               //Sign: Gemini

    ZodiacReader read19( june19 );                                      //One day before exact end date of Gemini

    assert( read19.stringifySign( read19.checkSign() ) == "Gemini" );   //Cusp: Cancer

    assert( read19.checkSign( ) == ZodiacReader::GEMINI );

    assert( read19.onCusp( ) );

    assert( read19.cuspSign( ) == ZodiacReader::CANCER );

    assert( read19.stringifySign( read19.cuspSign() ) == "Cancer" );

    Date june20( 6, 20 );                                               //Sign: Gemini

    ZodiacReader read20( june20 );                                      //On exact end date for Gemini, one day before exact start date for

    assert( read20.stringifySign( read20.checkSign() ) == "Gemini" );                                                               //Cancer

    assert( read20.checkSign( ) == ZodiacReader::GEMINI );              //Cusp: Cancer

    assert( read20.onCusp( ) );

    assert( read20.cuspSign( ) == ZodiacReader::CANCER );

    assert( read20.stringifySign( read20.cuspSign() ) == "Cancer" );

    //////////////////////////////////////////////////

    //      CANCER             JUNE 21 - JULY 22    //

    //////////////////////////////////////////////////

    Date june21( 6, 21 );                                               //Sign: Cancer

    ZodiacReader read21( june21 );                                      //On exact start date for Cancer, on day after exact end date for Gemini

    assert( read21.stringifySign( read21.checkSign() ) == "Cancer" );   //Cusp: Gemini

    assert( read21.checkSign( ) == ZodiacReader::CANCER );

    assert( read21.onCusp( ) );

    assert( read21.cuspSign( ) == ZodiacReader::GEMINI );

    assert( read21.stringifySign( read21.cuspSign() ) == "Gemini" );

    Date june22( 6, 22 );                                               //Sign: Cancer

    ZodiacReader read22( june22 );                                      //One day after exact start date

    assert( read22.stringifySign( read22.checkSign() ) == "Cancer" );   //Cusp: Gemini

    assert( read22.checkSign( ) == ZodiacReader::CANCER );

    assert( read22.onCusp( ) );

    assert( read22.cuspSign( ) == ZodiacReader::GEMINI );

    assert( read22.stringifySign( read22.cuspSign() ) == "Gemini" );

    Date june26( 6, 26 );

    ZodiacReader genTest11( june26 );

    assert( genTest11.stringifySign( genTest11.checkSign() ) == "Cancer" );

    assert( genTest11.checkSign( ) == ZodiacReader::CANCER );

    assert( !genTest11.onCusp( ) );

    Date june30( 6, 30 );                                               //End of month returns correct sign

    ZodiacReader read23( june30 );

    assert( read23.stringifySign( read23.checkSign() ) == "Cancer" );

    assert( read23.checkSign( ) == ZodiacReader::CANCER );

    assert( !read23.onCusp( ) );

    Date july1( 7, 1 );                                                 //Beginning of month returns correct sign

    ZodiacReader read24( july1 );

    assert( read24.stringifySign( read24.checkSign() ) == "Cancer" );

    assert( read24.checkSign( ) == ZodiacReader::CANCER );

    assert( !read24.onCusp( ) );

    Date july13( 7, 13 );

    ZodiacReader genTest12( july13 );

    assert( genTest12.stringifySign( genTest12.checkSign() ) == "Cancer" );

    assert( genTest12.checkSign( ) == ZodiacReader::CANCER );

    assert( !genTest12.onCusp( ) );

    Date july21( 7, 21 );                                               //Sign: Cancer

    ZodiacReader read25( july21 );                                      //One day before exact end date

    assert( read25.stringifySign( read25.checkSign() ) == "Cancer" );   //Cusp: Leo

    assert( read25.checkSign( ) == ZodiacReader::CANCER );

    assert( read25.onCusp( ) );

    assert( read25.cuspSign( ) == ZodiacReader::LEO);

    assert( read25.stringifySign( read25.cuspSign() ) == "Leo" );

    Date july22( 7, 22 );                                               //Sign: Cancer

    ZodiacReader read26( july22 );                                      //On exact end date for Cancer, one day before exact start date for Leo

    assert( read26.stringifySign( read26.checkSign() ) == "Cancer" );   //Cusp: Leo

    assert( read26.checkSign( ) == ZodiacReader::CANCER );

    assert( read26.onCusp( ) );

    assert( read26.cuspSign( ) == ZodiacReader::LEO);

    assert( read26.stringifySign( read26.cuspSign() ) == "Leo" );

    //////////////////////////////////////////////////

    //      LEO            JULY 23 - AUGUST 22      //

    //////////////////////////////////////////////////

    Date july23( 7, 23 );                                               //Sign: Leo

    ZodiacReader read27( july23 );                                      //On exact start date for Leo, one day after exact end date for Cancer

    assert( read27.stringifySign( read27.checkSign() ) == "Leo" );      //Cusp: Cancer

    assert( read27.checkSign( ) == ZodiacReader::LEO );

    assert( read27.onCusp( ) );

    assert( read27.cuspSign( ) == ZodiacReader::CANCER);

    assert( read27.stringifySign( read27.cuspSign() ) == "Cancer" );

    Date july24( 7, 24 );                                               //Sign: Leo

    ZodiacReader read28( july24 );                                      //One day after exact start date for Leo

    assert( read28.stringifySign( read28.checkSign() ) == "Leo" );      //Cusp: Cancer

    assert( read28.checkSign( ) == ZodiacReader::LEO );

    assert( read28.onCusp( ) );

    assert( read28.cuspSign( ) == ZodiacReader::CANCER);

    assert( read28.stringifySign( read28.cuspSign() ) == "Cancer" );

    Date july30( 7, 30 );

    ZodiacReader genTest13( july30 );

    assert( genTest13.stringifySign( genTest13.checkSign() ) == "Leo" );

    assert( genTest13.checkSign( ) == ZodiacReader::LEO );

    assert( !genTest13.onCusp( ) );

    Date july31( 7, 31 );                                               //End of month returns correct sign

    ZodiacReader read29( july31 );

    assert( read29.stringifySign( read29.checkSign() ) == "Leo" );

    assert( read29.checkSign( ) == ZodiacReader::LEO );

    assert( !read29.onCusp( ) );

    Date august1( 8, 1 );                                               //Beginning of month returns correct sign

    ZodiacReader read30( august1 );

    assert( read30.stringifySign( read30.checkSign() ) == "Leo" );

    assert( read30.checkSign( ) == ZodiacReader::LEO );

    assert( !read30.onCusp( ) );

    Date august14( 8, 14 );

    ZodiacReader genTest14( august14 );

    assert( genTest14.stringifySign( genTest14.checkSign() ) == "Leo" );

    assert( genTest14.checkSign( ) == ZodiacReader::LEO );

    assert( !genTest14.onCusp( ) );

    Date august21( 8, 21 );                                             //Sign: Leo

    ZodiacReader read31( august21 );                                    //One day before exact end date for Leo

    assert( read31.stringifySign( read31.checkSign() ) == "Leo" );      //Cusp: Virgo

    assert( read31.checkSign( ) == ZodiacReader::LEO );

    assert( read31.onCusp( ) );

    assert( read31.cuspSign( ) == ZodiacReader::VIRGO);

    assert( read31.stringifySign( read31.cuspSign() ) == "Virgo" );

    Date august22( 8, 22 );                                             //Sign: Leo

    ZodiacReader read32( august22 );                                    //On exact end date for Leo, one day before exact start date for Virgo

    assert( read32.stringifySign( read32.checkSign() ) == "Leo" );      //Cusp: Virgo

    assert( read32.checkSign( ) == ZodiacReader::LEO );

    assert( read32.onCusp( ) );

    assert( read32.cuspSign( ) == ZodiacReader::VIRGO);

    assert( read32.stringifySign( read32.cuspSign() ) == "Virgo" );

    //////////////////////////////////////////////////

    //      VIRGO     AUGUST 23 - SEPTEMBER 22      //

    //////////////////////////////////////////////////

    Date august23( 8, 23 );                                             //Sign: Virgo

    ZodiacReader read33( august23 );                                    //On exact start date for Virgo, one day after exact end date for Leo

    assert( read33.stringifySign( read33.checkSign() ) == "Virgo" );    //Cusp: Leo

    assert( read33.checkSign( ) == ZodiacReader::VIRGO);

    assert( read33.onCusp( ) );

    assert( read33.cuspSign( ) == ZodiacReader::LEO);

    assert( read33.stringifySign( read33.cuspSign() ) == "Leo" );

    Date august24( 8, 24 );                                             //Sign: Virgo

    ZodiacReader read34( august24 );                                    //One day after exact start date for Virgo

    assert( read34.stringifySign( read34.checkSign() ) == "Virgo" );    //Cusp: Leo

    assert( read34.checkSign( ) == ZodiacReader::VIRGO);

    assert( read34.onCusp( ) );

    assert( read34.cuspSign( ) == ZodiacReader::LEO);

    assert( read34.stringifySign( read34.cuspSign() ) == "Leo" );

    Date august29( 8, 29 );

    ZodiacReader genTest15( august29 );

    assert( genTest15.stringifySign( genTest15.checkSign() ) == "Virgo" );

    assert( genTest15.checkSign( ) == ZodiacReader::VIRGO);

    assert( !genTest15.onCusp( ) );

    Date august31( 8, 31 );                                             //End of month returns correct sign

    ZodiacReader read35( august31 );

    assert( read35.stringifySign( read35.checkSign() ) == "Virgo" );

    assert( read35.checkSign( ) == ZodiacReader::VIRGO);

    assert( !read35.onCusp( ) );

    Date sept1( 9, 1 );                                                 //Beginning of month returns correct sign

    ZodiacReader read36( sept1 );

    assert( read36.stringifySign( read36.checkSign() ) == "Virgo" );

    assert( read36.checkSign( ) == ZodiacReader::VIRGO);

    assert( !read36.onCusp( ) );

    Date sept12( 9, 12 );

    ZodiacReader genTest16( sept1 );

    assert( genTest16.stringifySign( genTest16.checkSign() ) == "Virgo" );

    assert( genTest16.checkSign( ) == ZodiacReader::VIRGO);

    assert( !genTest16.onCusp( ) );

    Date sept21( 9, 21 );                                               //Sign: Virgo

    ZodiacReader read37( sept21 );                                      //One day before exact end date for Virgo

    assert( read37.stringifySign( read37.checkSign() ) == "Virgo" );    //Cusp: Libra

    assert( read37.checkSign( ) == ZodiacReader::VIRGO);

    assert( read37.onCusp( ) );

    assert( read37.cuspSign( ) == ZodiacReader::LIBRA);

    assert( read37.stringifySign( read37.cuspSign() ) == "Libra" );

    Date sept22( 9, 22 );                                               //Sign: Virgo

    ZodiacReader read38( sept22 );                                      //On exact end date for Virgo, one day before exact start date for Libra

    assert( read38.stringifySign( read38.checkSign() ) == "Virgo" );    //Cusp: Libra

    assert( read38.checkSign( ) == ZodiacReader::VIRGO);

    assert( read38.onCusp( ) );

    assert( read38.cuspSign( ) == ZodiacReader::LIBRA);

    assert( read38.stringifySign( read38.cuspSign() ) == "Libra" );

    //////////////////////////////////////////////////

    //      LIBRA     SEPTEMBER 23 - OCTOBER 22     //

    //////////////////////////////////////////////////

    Date sept23( 9, 23 );                                               //Sign: Libra

    ZodiacReader read39( sept23 );                                      //On exact start date for Libra, one day after exact end date for Virgp

    assert( read39.stringifySign( read39.checkSign() ) == "Libra" );    //Cusp: Virgo

    assert( read39.checkSign( ) == ZodiacReader::LIBRA);

    assert( read39.onCusp( ) );

    assert( read39.cuspSign( ) == ZodiacReader::VIRGO);

    assert( read39.stringifySign( read39.cuspSign() ) == "Virgo" );

    Date sept24( 9, 24 );                                               //Sign: Libra

    ZodiacReader read40( sept24 );                                      //One day after exact start date for Libra

    assert( read40.stringifySign( read40.checkSign() ) == "Libra" );    //Cusp: Virgo

    assert( read40.checkSign( ) == ZodiacReader::LIBRA);

    assert( read40.onCusp( ) );

    assert( read40.cuspSign( ) == ZodiacReader::VIRGO);

    assert( read40.stringifySign( read40.cuspSign() ) == "Virgo" );

    Date sept27( 9, 27 );

    ZodiacReader genTest17( sept27 );

    assert( genTest17.stringifySign( genTest17.checkSign() ) == "Libra" );

    assert( genTest17.checkSign( ) == ZodiacReader::LIBRA);

    assert( !genTest17.onCusp( ) );

    Date sept30( 9, 30 );                                               //End of month returns correct sign

    ZodiacReader read41( sept30 );

    assert( read41.stringifySign( read41.checkSign() ) == "Libra" );

    assert( read41.checkSign( ) == ZodiacReader::LIBRA);

    assert( !read41.onCusp( ) );

    Date oct1( 10, 1 );                                                 //Beginning of month returns correct sign

    ZodiacReader read42( oct1 );

    assert( read42.stringifySign( read42.checkSign() ) == "Libra" );

    assert( read42.checkSign( ) == ZodiacReader::LIBRA);

    assert( !read42.onCusp( ) );

    Date oct14( 10, 14 );

    ZodiacReader genTest18( oct14 );

    assert( genTest18.stringifySign( genTest18.checkSign() ) == "Libra" );

    assert( genTest18.checkSign( ) == ZodiacReader::LIBRA);

    assert( !genTest18.onCusp( ) );

    Date oct21( 10, 21 );                                               //Sign: Libra

    ZodiacReader read43( oct21 );                                       //One day before exact end date

    assert( read43.stringifySign( read43.checkSign() ) == "Libra" );    //Cusp: Scorpio

    assert( read43.checkSign( ) == ZodiacReader::LIBRA);

    assert( read43.onCusp( ) );

    assert( read43.cuspSign( ) == ZodiacReader::SCORPIO);

    assert( read43.stringifySign( read43.cuspSign() ) == "Scorpio" );

    Date oct22( 10, 22 );                                              //Sign: Libra

    ZodiacReader read44( oct22 );                                      //On exact end date for Libra, one day before exact start date for Scorpio

    assert( read44.stringifySign( read44.checkSign() ) == "Libra" );   //Cusp: Scorpio

    assert( read44.checkSign( ) == ZodiacReader::LIBRA);

    assert( read44.onCusp( ) );

    assert( read44.cuspSign( ) == ZodiacReader::SCORPIO);

    assert( read44.stringifySign( read44.cuspSign() ) == "Scorpio" );

    //////////////////////////////////////////////////

    //      SCORPIO     OCTOBER 23 - NOVEMBER 21    //

    //////////////////////////////////////////////////

    Date oct23( 10, 23 );                                               //Sign: Scorpio

    ZodiacReader read45( oct23 );                                       //On exact start date for Scorpio, one dat after exact end date for Libra

    assert( read45.stringifySign( read45.checkSign() ) == "Scorpio" );  //Cusp: Libra

    assert( read45.checkSign( ) == ZodiacReader::SCORPIO);

    assert( read45.onCusp( ) );

    assert( read45.cuspSign( ) == ZodiacReader::LIBRA);

    assert( read45.stringifySign( read45.cuspSign() ) == "Libra" );

    Date oct24( 10, 24 );                                               //Sign: Scorpio

    ZodiacReader read46( oct24 );                                       //One day after exact start date

    assert( read46.stringifySign( read46.checkSign() ) == "Scorpio" );  //Cusp: Libra

    assert( read46.checkSign( ) == ZodiacReader::SCORPIO);

    assert( read46.onCusp( ) );

    assert( read46.cuspSign( ) == ZodiacReader::LIBRA);

    assert( read46.stringifySign( read46.cuspSign() ) == "Libra" );

    Date oct26( 10, 26 );

    ZodiacReader genTest19( oct26 );

    assert( genTest19.stringifySign( genTest19.checkSign() ) == "Scorpio" );

    assert( genTest19.checkSign( ) == ZodiacReader::SCORPIO);

    assert( !genTest19.onCusp( ) );

    Date oct31( 10, 31 );                                               //End of month returns correct sign

    ZodiacReader read47( oct31 );

    assert( read47.stringifySign( read47.checkSign() ) == "Scorpio" );

    assert( read47.checkSign( ) == ZodiacReader::SCORPIO);

    assert( !read47.onCusp( ) );

    Date nov1( 11, 1 );                                                 //Beginning of month returns correct sign

    ZodiacReader read48( nov1 );

    assert( read48.stringifySign( read48.checkSign() ) == "Scorpio" );

    assert( read48.checkSign( ) == ZodiacReader::SCORPIO);

    assert( !read48.onCusp( ) );

    Date nov15( 11, 15 );

    ZodiacReader genTest20( nov15 );

    assert( genTest20.stringifySign( genTest20.checkSign() ) == "Scorpio" );

    assert( genTest20.checkSign( ) == ZodiacReader::SCORPIO);

    assert( !genTest20.onCusp( ) );

    Date nov20( 11, 20 );                                               //Sign: Scorpio

    ZodiacReader read49( nov20 );                                       //One day before exact end date

    assert( read49.stringifySign( read49.checkSign() ) == "Scorpio" );  //Cusp: Sagittarius

    assert( read49.checkSign( ) == ZodiacReader::SCORPIO);

    assert( read49.onCusp( ) );

    assert( read49.cuspSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read49.stringifySign( read49.cuspSign() ) == "Sagittarius" );

    Date nov21( 11, 21 );                                        //Sign: Scorpio

    ZodiacReader read50( nov21 );                                //On exact end date of Scorpio, one day before exact start date for Sagittarius

    assert( read50.stringifySign( read50.checkSign() ) == "Scorpio" );  //Cusp: Sagittarius

    assert( read50.checkSign( ) == ZodiacReader::SCORPIO);

    assert( read50.onCusp( ) );

    assert( read50.cuspSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read50.stringifySign( read50.cuspSign() ) == "Sagittarius" );

    //////////////////////////////////////////////////

    // SAGITTARIUS     NOVEMBER 22 - DECEMBER 21    //

    //////////////////////////////////////////////////

    Date nov22( 11, 22 );                                       //Sign: Sagittarius

    ZodiacReader read51( nov22 );                               //On exact start date for Sagittarius, one day after exact end date for Scorpio

    assert( read51.stringifySign( read51.checkSign() ) == "Sagittarius" );  //Cusp: Scorpio

    assert( read51.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read51.onCusp( ) );

    assert( read51.cuspSign( ) == ZodiacReader::SCORPIO);

    assert( read51.stringifySign( read51.cuspSign() ) == "Scorpio" );

    Date nov23( 11, 23 );                                       //Sign: Sagittarius

    ZodiacReader read60( nov23 );                               //One day after exact start date for Sagittarius

    assert( read60.stringifySign( read60.checkSign() ) == "Sagittarius" );  //Cusp: Scorpio

    assert( read60.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read60.onCusp( ) );

    assert( read60.cuspSign( ) == ZodiacReader::SCORPIO);

    assert( read60.stringifySign( read60.cuspSign() ) == "Scorpio" );

    Date nov27( 11, 27 );

    ZodiacReader genTest21( nov27 );

    assert( genTest21.stringifySign( genTest21.checkSign() ) == "Sagittarius" );

    assert( genTest21.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( !genTest21.onCusp( ) );

    Date nov30( 11, 30 );                                       //End of month returns correct sign

    ZodiacReader read52( nov30 );

    assert( read52.stringifySign( read52.checkSign() ) == "Sagittarius" );

    assert( read52.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( !read52.onCusp( ) );

    Date dec1( 12, 1 );                                         //Beginning of month returns correct sign

    ZodiacReader read53( dec1 );

    assert( read53.stringifySign( read53.checkSign() ) == "Sagittarius" );

    assert( read53.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( !read53.onCusp( ) );

    Date dec12( 12, 12 );

    ZodiacReader genTest22( dec12 );

    assert( genTest22.stringifySign( genTest22.checkSign() ) == "Sagittarius" );

    assert( genTest22.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( !genTest22.onCusp( ) );

    Date dec20( 12, 20 );                                       //Sign: Sagittarius

    ZodiacReader read54( dec20 );                               //One day before exact end date for Sagittarius

    assert( read54.stringifySign( read54.checkSign() ) == "Sagittarius" );  //Cusp: Capricorn

    assert( read54.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read54.onCusp( ) );

    assert( read54.cuspSign( ) == ZodiacReader::CAPRICORN);

    assert( read54.stringifySign( read54.cuspSign() ) == "Capricorn" );

    Date dec21( 12, 21 );                                      //Sign: Sagittarius

    ZodiacReader read55( dec21 );                             //On exact end date for Sagittarius, one day before exact start date for Capricorn

    assert( read55.stringifySign( read55.checkSign() ) == "Sagittarius" ); //Cusp: Capricorn

    assert( read55.checkSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read55.onCusp( ) );

    assert( read55.cuspSign( ) == ZodiacReader::CAPRICORN);

    assert( read55.stringifySign( read55.cuspSign() ) == "Capricorn" );

    //////////////////////////////////////////////////

    // CAPRICORN       DECEMBER 22 - JANUARY 19     //

    //////////////////////////////////////////////////

    Date dec22( 12, 22 );                                       //Sign: Capricorn

    ZodiacReader read56( dec22 );                               //On exact start date for Capricorn, one day after exact end date for Sagittarius

    assert( read56.stringifySign( read56.checkSign() ) == "Capricorn" );    //Cusp: Sagittarius

    assert( read56.checkSign( ) == ZodiacReader::CAPRICORN);

    assert( read56.onCusp( ) );

    assert( read56.cuspSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read56.stringifySign( read56.cuspSign() ) == "Sagittarius" );

    Date dec23( 12, 23 );                                       //Sign: Capricorn

    ZodiacReader read57( dec23 );                               //One day after exact start date for Capricorn

    assert( read57.stringifySign( read57.checkSign() ) == "Capricorn" );    //Cusp: Sagittarius

    assert( read57.checkSign( ) == ZodiacReader::CAPRICORN);

    assert( read57.onCusp( ) );

    assert( read57.cuspSign( ) == ZodiacReader::SAGITTARIUS);

    assert( read57.stringifySign( read57.cuspSign() ) == "Sagittarius" );

    Date dec25( 12, 25 );

    ZodiacReader genTest23( dec25 );

    assert( genTest23.stringifySign( genTest23.checkSign() ) == "Capricorn" );

    assert( genTest23.checkSign( ) == ZodiacReader::CAPRICORN);

    assert( !genTest23.onCusp( ) );

    Date dec31( 12, 31 );                                       //End of month returns correct sign

    ZodiacReader read58( dec31 );

    assert( read58.stringifySign( read58.checkSign() ) == "Capricorn" );

    assert( read58.checkSign( ) == ZodiacReader::CAPRICORN);

    assert( !read58.onCusp( ) );

    Date jan1( 1, 1 );                                                      //Beginning of month returns correct sign

    ZodiacReader reader( jan1 );                                            //example test case on spec

    assert( reader.stringifySign( reader.checkSign() ) == "Capricorn" );

    assert( reader.checkSign( ) == ZodiacReader::CAPRICORN );

    assert( !reader.onCusp( ) );

    Date jan14( 1, 14 );

    ZodiacReader genTest24( jan14 );

    assert( genTest24.stringifySign( genTest24.checkSign() ) == "Capricorn" );

    assert( genTest24.checkSign( ) == ZodiacReader::CAPRICORN );

    assert( !genTest24.onCusp( ) );

    Date jan18( 1, 18 );                                                    //Sign: Capricorn

    ZodiacReader read59( jan18 );                                           //One day before exact end date

    assert( read59.stringifySign( read59.checkSign() ) == "Capricorn" );    //Cusp: Aquarius

    assert( read59.checkSign( ) == ZodiacReader::CAPRICORN );

    assert( read59.onCusp( ) );

    assert( read59.cuspSign( ) == ZodiacReader::AQUARIUS );

    assert( read59.stringifySign( read59.cuspSign() ) == "Aquarius" );

    Date jan19( 1, 19 );                                         //Sign: Capricorn

    ZodiacReader reader1( jan19 );                               //On exact end date for Capricorn, one day before exact start date for Aquarius

    assert( reader1.stringifySign( reader1.checkSign() ) == "Capricorn" );  //Cusp: Aquarius

    assert( reader1.checkSign( ) == ZodiacReader::CAPRICORN );

    assert( reader1.onCusp( ) );

    assert( reader1.cuspSign( ) == ZodiacReader::AQUARIUS );

    assert( reader1.stringifySign( reader1.cuspSign() ) == "Aquarius" );

    //////////////////////////////////////////////////

    // AQUARIUS       JANUARY 20 - FEBRUARY 18      //

    //////////////////////////////////////////////////

    Date jan20( 1, 20 );                                           //Sign: Aquarius

    ZodiacReader reader2( jan20 );                                 //On exact start date for Aquarius, one day after exact end date for Capricorn

    assert( reader2.stringifySign( reader2.checkSign() ) == "Aquarius" );   //Cusp: Capricorn

    assert( reader2.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( reader2.onCusp( ) );

    assert( reader2.cuspSign( ) == ZodiacReader::CAPRICORN );

    assert( reader2.stringifySign( reader2.cuspSign() ) == "Capricorn" );

    Date jan21( 1, 21 );                                                    //Sign: Aquarius

    ZodiacReader reader3( jan21 );                                          //One day after exact start date for Aquarius

    assert( reader3.stringifySign( reader3.checkSign() ) == "Aquarius" );   //Cusp: Capricorn

    assert( reader3.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( reader3.onCusp( ) );

    assert( reader3.cuspSign( ) == ZodiacReader::CAPRICORN );

    assert( reader3.stringifySign( reader3.cuspSign() ) == "Capricorn" );

    Date jan22( 1, 22 );

    ZodiacReader genTest( jan22 );

    assert( genTest.stringifySign( genTest.checkSign() ) == "Aquarius" );

    assert( genTest.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !genTest.onCusp( ) );

    Date jan26( 1, 26 );

    ZodiacReader genTest1( jan22 );

    assert( genTest1.stringifySign( genTest1.checkSign() ) == "Aquarius" );

    assert( genTest1.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !genTest1.onCusp( ) );

    Date jan30( 1, 30 );

    ZodiacReader genTest2( jan30 );

    assert( genTest2.stringifySign( genTest2.checkSign() ) == "Aquarius" );

    assert( genTest2.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !genTest2.onCusp( ) );

    Date jan31( 1, 31 );                                                    //End of month returns correct sign

    ZodiacReader reader4( jan31 );

    assert( reader4.stringifySign( reader2.checkSign() ) == "Aquarius" );

    assert( reader4.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !reader4.onCusp( ) );

    Date feb1( 2, 1 );                                                      //Beginning of month returns correct sign

    ZodiacReader reader5( feb1 );

    assert( reader5.stringifySign( reader5.checkSign() ) == "Aquarius" );

    assert( reader5.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !reader5.onCusp( ) );

    Date feb14( 2, 14 );

    ZodiacReader genTest3( feb14 );

    assert( genTest3.stringifySign( genTest3.checkSign() ) == "Aquarius" );

    assert( genTest3.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( !genTest3.onCusp( ) );

    Date feb17( 2, 17 );                                                    //Sign: Aquarius

    ZodiacReader reader6( feb17 );                                          //One day before exact end date for Aquarius

    assert( reader6.stringifySign( reader6.checkSign() ) == "Aquarius" );   //Cusp: Pisces

    assert( reader6.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( reader6.onCusp( ) );

    assert( reader6.cuspSign( ) == ZodiacReader::PISCES );

    assert( reader6.stringifySign( reader6.cuspSign() ) == "Pisces" );

    Date feb18( 2, 18 );                                            //Sign: Aquarius

    ZodiacReader reader7( feb18 );                                  //On exact end date for Aquarius, one day before exact start date for Pisces

    assert( reader7.stringifySign( reader7.checkSign() ) == "Aquarius" );   //Cusp: Pisces

    assert( reader7.checkSign( ) == ZodiacReader::AQUARIUS );

    assert( reader7.onCusp( ) );

    assert( reader7.cuspSign( ) == ZodiacReader::PISCES );

    assert( reader7.stringifySign( reader7.cuspSign() ) == "Pisces" );

    //////////////////////////////////////////////////

    // PISCES           FEBRUARY 19 - MARCH 20      //

    //////////////////////////////////////////////////

    Date feb19( 2, 19 );                                            //Sign: Pisces

    ZodiacReader reader8( feb19 );                                  //On exact start date for Pisces, one day after exact end date for Aquarius

    assert( reader8.stringifySign( reader8.checkSign() ) == "Pisces" ); //Cusp: Aquarius

    assert( reader8.checkSign( ) == ZodiacReader::PISCES );

    assert( reader8.onCusp( ) );

    assert( reader8.cuspSign( ) == ZodiacReader::AQUARIUS );

    assert( reader8.stringifySign( reader8.cuspSign() ) == "Aquarius" );

    Date feb20( 2, 20 );                                                //Sign: Pisces

    ZodiacReader reader9( feb20 );                                      //One day after exact start date for Pisces

    assert( reader9.stringifySign( reader9.checkSign() ) == "Pisces" ); //Cusp: Aquarius

    assert( reader9.checkSign( ) == ZodiacReader::PISCES );

    assert( reader9.onCusp( ) );

    assert( reader9.cuspSign( ) == ZodiacReader::AQUARIUS );

    assert( reader9.stringifySign( reader9.cuspSign() ) == "Aquarius" );

    Date feb25( 2, 25 );

    ZodiacReader genTest4( feb25 );

    assert( genTest4.stringifySign( genTest4.checkSign() ) == "Pisces" );

    assert( genTest4.checkSign( ) == ZodiacReader::PISCES );

    assert( !genTest4.onCusp( ) );

    Date feb28( 2, 28 );                                                //End of month returns correct sign

    ZodiacReader reader10( feb28 );

    assert( reader10.stringifySign( reader10.checkSign() ) == "Pisces" );

    assert( reader10.checkSign( ) == ZodiacReader::PISCES );

    assert( !reader10.onCusp( ) );

    Date mar1( 3, 1 );                                                  //Beginning of month returns correct sign

    ZodiacReader reader11( mar1 );

    assert( reader11.stringifySign( reader11.checkSign() ) == "Pisces" );

    assert( reader11.checkSign( ) == ZodiacReader::PISCES );

    assert( !reader11.onCusp( ) );

    Date mar9( 3, 9 );

    ZodiacReader genTest5( mar9 );

    assert( genTest5.stringifySign( genTest5.checkSign() ) == "Pisces" );

    assert( genTest5.checkSign( ) == ZodiacReader::PISCES );

    assert( !genTest5.onCusp( ) );

    Date mar13( 3, 13 );

    ZodiacReader genTest6( mar9 );

    assert( genTest6.stringifySign( genTest6.checkSign() ) == "Pisces" );

    assert( genTest6.checkSign( ) == ZodiacReader::PISCES );

    assert( !genTest6.onCusp( ) );

    Date mar19( 3, 19 );                                                    //Sign: Pisces

    ZodiacReader reader12( mar19 );                                         //One day before exact end date for Pisces

    assert( reader12.stringifySign( reader12.checkSign() ) == "Pisces" );   //Cusp: Aries

    assert( reader12.checkSign( ) == ZodiacReader::PISCES );

    assert( reader12.onCusp( ) );

    assert( reader12.cuspSign( ) == ZodiacReader::ARIES );

    assert( reader12.stringifySign( reader12.cuspSign() ) == "Aries" );

    Date mar20( 3, 20 );                                                //Sign: Pisces

    ZodiacReader reader13( mar20 );                                    //On exact end date for Pisces, one day before exact start date for Aries

    assert( reader13.stringifySign( reader13.checkSign() ) == "Pisces" );   //Cusp: Aries

    assert( reader13.checkSign( ) == ZodiacReader::PISCES );

    assert( reader13.onCusp( ) );

    assert( reader13.cuspSign( ) == ZodiacReader::ARIES );

    assert( reader13.stringifySign( reader13.cuspSign() ) == "Aries" );