Classes, member functions, operators

1. Create a minimal class 'Myclass' that can hold one variable, and that has member functions s that this main() code fragment works.

   ```
   int main() {
       Myclass a;
       a.set("I am happy");
       a.print()
       return 0;
   }
   ```

   The program will print this after finished running:

   ```
   Stored: I am happy
   I am happy
   ```

2. Create a second class that inherits from the first one and that has two additional string variable, that print 50% of the time:

   ```
   I am happy
   when it rains.
   ```

   or

   ```
   I am happy
   when the sun shines.
   ```

   using the print() of the inherited class and new or extended print() statement.

3. Finish the 'Rational class project', decide how you want to test all the new methods. The project was written on a Mac using netbeans 7.3, use 'open project'. Netbeans will complain several times that this project was from a newer version and that it came from a Mac (if you run windows); ignore all these warnings and accept and let it compile. My first build succeeded but did not run, my second build run successfully. Do not edit any code before you have not checked that it runs successfully, and then change code and test using smallest units, for example: add an operator==() and test immediately by using an *if-statement* in main.cpp [! remember, we have now three files: main.ccp, rational.cpp, rational.h], the statement could be like this:

   ```
   if (a==b) {
   cout << "both fractions are the same" << endl;
   } else {
   cout << "fractions are different" << endl;
   }
   ```